

(11)Publication number : 07-271355
(43)Date of publication of application : 20.10.1995

G10F 1/02

(72)Inventor : FUJIWARA YUJI
KAWABATA TARO
ISOZAKI YOSHIMASA
OBA YASUHIKO

26.10.2000

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-271355

(43) 公開日 平成7年(1995)10月20日

(51) Int.Cl.⁶

G 1 0 F 1/02

識別記号

B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 7 F D (全 29 頁)

(21) 出願番号 特願平6-79604

(22) 出願日 平成6年(1994)3月25日

(71) 出願人 000004075

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(72) 発明者 藤原 祐二

静岡県浜松市中沢町10番1号 ヤマハ株式会社内

(72) 発明者 川端 太郎

静岡県浜松市中沢町10番1号 ヤマハ株式会社内

(72) 発明者 磯崎 善政

静岡県浜松市中沢町10番1号 ヤマハ株式会社内

(74) 代理人 弁理士 川▲崎▼ 研二 (外1名)

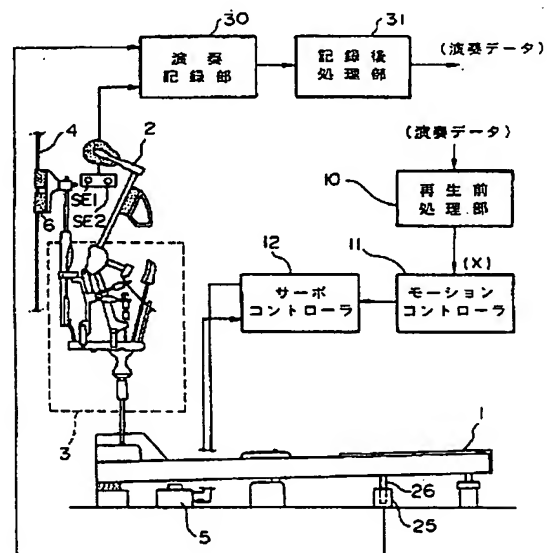
最終頁に続く

(54) 【発明の名称】 自動ピアノおよび演奏データ加工装置

(57) 【要約】

【目的】 打弦速度や演奏の微妙なニュアンスを正確に再現する。

【構成】 再生前処理部10は、記録メディアから供給される演奏データに基づいて、リファレンス時刻 t_r 、リファレンス速度 V_r を求め、さらに、加速度の推定が必要な場合はこれを演算する。ここで、演奏データにおいて鍵あるいはハンマーのイベントが抜けている場合は、抜けたイベントを適宜修復する。そして、軌道の種類(直線、放物線等)に応じて鍵軌道を生成する。モーションコントローラ11は、再生前処理部10が作成した軌道データに基づいて、ソレノイド5のプランジャの動きを示すデータを作成し、このデータに基づいてサーボコントローラ12がソレノイド5の励磁電流をサーボ制御する。この結果、鍵1がリファレンスポイント X_r に達したときの速度は、記録時の打弦速度に対応した速度になり、ハンマー2は記録時と同じ速度で打弦を行う。



【特許請求の範囲】

【請求項 1】 複数のイベントデータから成る演奏データが供給されると、これら各イベントデータ相互の関係に基づいて、欠落したイベントデータが存在するかどうかを判定する判定手段と、

前記判定手段において欠落したイベントデータが存在すると判定されると、前記演奏データに新たなイベントデータを適宜追加するイベント修復手段とを具備することを特徴とする演奏データ加工装置。

【請求項 2】 各々が時刻情報を含む複数のイベントデータから成る演奏データが供給されると、前記各イベントデータに所定の処理を施して出力するイベントデータ加工手段と、

前記イベントデータ加工手段において各イベントデータに費やされる処理時間と、遅延時間との合計が所定時間になるように、前記各イベントデータを遅延させる遅延手段とを具備することを特徴とする演奏データ加工装置。

【請求項 3】 複数のイベントデータから成る演奏データが供給されると、これら各イベントデータを適宜遅延して出力する遅延手段と、

前記各イベントデータに係るイベントの実行タイミングに基づいて、前記遅延手段における各イベントデータの遅延時間を各々設定する遅延時間設定手段とを具備することを特徴とする演奏データ加工装置。

【請求項 4】 前記演奏データは、押鍵イベント、ハンマーイベントおよび離鍵イベントに係るイベントデータを含むものであり、

前記判定手段は、押鍵イベント、ハンマーイベント、離鍵イベントの順序で前記各イベントデータが順次供給されなかった場合に、欠落したイベントデータが存在すると判定することを特徴とする請求項 1 記載の演奏データ加工装置。

【請求項 5】 前記演奏データは、押鍵イベント、ハンマーイベントおよび離鍵イベントに係るイベントデータを含むものであり、

前記判定手段は、一の押鍵イベントと該押鍵イベントに対応するハンマーイベントとの時間間隔が所定時間以上である場合に他のハンマーイベントが欠落したものと判定するものであり、

前記イベント修復手段は、前記他のハンマーイベントに係るイベントデータを前記演奏データに追加するとともに、前記他のハンマーイベントにおける打弦速度を、打弦時に発音が行われない程度の速度に設定することを特徴とする請求項 1 記載の演奏データ加工装置。

【請求項 6】 前記遅延時間設定手段は、供給された前記演奏データに基づいて、前記実行タイミングを生成することを特徴とする請求項 3 記載の演奏データ加工装置。

【請求項 7】 所定の鍵について押鍵イベントと打弦イベ

ントとを検出するイベント検出手段と、

前記押鍵イベントと前記打弦イベントとを記憶する記憶手段と、

前記記憶手段の内容を読み出し、前記押鍵イベントまたは前記打弦イベントのうち一方が欠落している場合は欠落しているイベントを追加し、その結果を演奏データとして出力するイベント修復手段と、

前記演奏データに基づいて鍵軌道を生成し、この鍵軌道に沿って前記鍵を駆動する鍵駆動手段とを具備することを特徴とする自動ピアノ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、例えば、自動的に鍵盤を駆動して演奏を行う自動ピアノおよびこの自動ピアノに用いて好適な演奏データ加工装置に関する。

【0002】

【従来の技術】従来の自動ピアノにおいては、各鍵および各ハンマーに対してセンサが設けられており、これらセンサによって検出したオン/オフイベント、押鍵速度またはハンマー速度等に基づいて演奏データを記録し、この演奏データに基づいて押鍵/離鍵時刻および押鍵速度を制御することにより演奏状態を再生することが可能である。また、自動ピアノにおいて検出した演奏データを、他の自動ピアノあるいは電子楽器等に直接供給することにより、複数箇所リアルタイムに生演奏を行うことも可能である。

【0003】また、これは公知技術ではないが、単に押鍵/離鍵時刻および押鍵速度等に基づいて各鍵を駆動したのでは演奏の微妙なニュアンスを再現することが困難であるため、リファレンス軌道（鍵軌道の目標値）を生成し連続的に各鍵の動きを駆動する技術が本出願人により提案されている（平成 5 年 12 月 17 日付の特許出願、整理番号 937013）。

【0004】

【発明が解決しようとする課題】ところで、上述した自動ピアノは、押鍵イベント（押鍵時刻、押鍵速度）、ハンマーイベント（打弦時刻、打弦速度）等に基づいてリファレンス軌道を生成するため、なんらかの理由により何れかのイベントが抜けた場合は、リファレンス軌道を生成することが困難であった。この発明は上述した事情に鑑みてなされたものであり、取得された限りの情報に基づいて正確な鍵の動作再生を行う自動ピアノを提供することを第 1 の目的としている。また、かかる鍵の動作再生をリアルタイムに実現するために、この自動ピアノに用いて好適な演奏データ加工装置を提供することを第 2 の目的としている。

【0005】

【課題を解決するための手段】上記課題を解決するため、請求項 1 に係る構成にあっては、複数のイベントデータから成る演奏データが供給されると、これら各イベ

ントデータ相互の関係に基づいて、欠落したイベントデータが存在するか否かを判定する判定手段と、前記判定手段において欠落したイベントデータが存在すると判定されると、前記演奏データに新たなイベントデータを適宜追加するイベント修復手段とを具備することを特徴としている。

【0006】また、請求項2に係る構成にあっては、各々が時刻情報を含む複数のイベントデータから成る演奏データが供給されると、前記各イベントデータに所定の処理を施して出力するイベントデータ加工手段と、前記イベントデータ加工手段において各イベントデータに費やされる処理時間と、遅延時間との合計が所定時間になるように、前記各イベントデータを遅延させる遅延手段とを具備することを特徴としている。

【0007】また、請求項3に係る構成にあっては、複数のイベントデータから成る演奏データが供給されると、これら各イベントデータを適宜遅延して出力する遅延手段と、前記各イベントデータに係るイベントの実行タイミングに基づいて、前記遅延手段における各イベントデータの遅延時間を各々設定する遅延時間設定手段とを具備することを特徴としている。

【0008】また、請求項4に係る構成にあっては、請求項1において、前記演奏データは、押鍵イベント、ハンマーイベントおよび離鍵イベントに係るイベントデータを含むものであり、前記判定手段は、押鍵イベント、ハンマーイベント、離鍵イベントの順序で前記各イベントデータが順次供給されなかった場合に、欠落したイベントデータが存在すると判定することを特徴としている。

【0009】また、請求項5に係る構成にあっては、請求項1において、前記演奏データは、押鍵イベント、ハンマーイベントおよび離鍵イベントに係るイベントデータを含むものであり、前記判定手段は、一の押鍵イベントと該押鍵イベントに対応するハンマーイベントとの時間間隔が所定時間以上である場合に他のハンマーイベントが欠落したものと判定するものであり、前記イベント修復手段は、前記他のハンマーイベントに係るイベントデータを前記演奏データに追加するとともに、前記他のハンマーイベントにおける打弦速度を、打弦時に発音が行われない程度の速度に設定することを特徴としている。

【0010】また、請求項6に係る構成にあっては、請求項3において、前記遅延時間設定手段は、供給された前記演奏データに基づいて、前記実行タイミングを生成することを特徴としている。

【0011】また、請求項7に係る構成にあっては、所定の鍵について押鍵イベントと打弦イベントとを検出するイベント検出手段と、前記押鍵イベントと前記打弦イベントとを記憶する記憶手段と、前記記憶手段の内容を讀出し、前記押鍵イベントまたは前記打弦イベントのう

ち一方が欠落している場合は欠落しているイベントを追加し、その結果を演奏データとして出力するイベント修復手段と、前記演奏データに基づいて鍵軌道を生成し、この鍵軌道に沿って前記鍵を駆動する鍵駆動手段とを具備することを特徴としている。

【0012】

【作用】請求項1に係る構成にあっては、判定手段は、複数のイベントデータ相互の関係に基づいて、欠落したイベントデータが存在するか否かを判定する。そして、欠落したイベントデータが存在すると判定されると、イベント修復手段は、演奏データに新たなイベントデータを適宜追加する。

【0013】また、請求項2に係る構成にあっては、イベントデータ加工手段は、各々が時刻情報を含む複数のイベントデータから成る演奏データが供給されると、これら各イベントデータに所定の処理を施して出力する。ここで、遅延手段は、イベントデータ加工手段において各イベントデータに費やされる処理時間と、遅延時間との合計が所定時間になるように、各イベントデータを遅延させるから、イベントデータ加工手段および遅延手段において費やされる合計時間は、該所定時間になる。

【0014】また、請求項3に係る構成にあっては、遅延手段は、複数のイベントデータから成る演奏データが供給されると、これら各イベントデータを適宜遅延して出力する。そして、遅延時間設定手段は各イベントデータに係るイベントの実行タイミングに基づいて、遅延手段における各イベントデータの遅延時間を各々設定するから、各イベントデータは、その実行タイミングに応じた時刻に出力されることになる。

【0015】また、請求項4に係る構成にあっては、押鍵イベント、ハンマーイベント、離鍵イベントの順序で各イベントデータが順次供給されなかった場合に、判定手段は、欠落したイベントデータが存在すると判定するから、各イベントの順序に基づいて抜けイベントの有無を判定することができる。

【0016】また、請求項5に係る構成にあっては、判定手段は、一の押鍵イベントと該押鍵イベントに対応するハンマーイベントとの時間間隔が所定時間以上である場合に他のハンマーイベントが欠落したものと判定し、イベント修復手段は該他のハンマーイベントに係るイベントデータを演奏データに追加するとともに、該他のハンマーイベントにおける打弦速度を、打弦時に発音が行われない程度の速度に設定する。これにより、ハンマーイベントによる発音を抑制しつつダンパ操作を行うことができる。

【0017】また、請求項6に係る構成にあっては、遅延時間設定手段は、供給された演奏データに基づいて実行タイミングを生成する。また、請求項7に係る構成にあっては、イベント検出手段が所定の鍵について押鍵イベントと打弦イベントとを検出すると、これらイベント

の内容が記憶手段に記憶される。イベント修復手段はこれら各イベントを読み出し、押鍵イベントまたは打弦イベントのうち一方が欠落している場合は欠落しているイベントを追加し、その結果を演奏データとして出力する。鍵駆動手段は、この演奏データに基づいて鍵を駆動するから、イベントの欠落があった場合であっても鍵を駆動することが可能になる。

【0018】

【実施例】

1. 第1実施例

A. 実施例の制御原理

①リファレンスポイント

図20は、一般的な自動ピアノの要部の構成を示す断面図である。図に示すように、自動ピアノにおいては、鍵1と、鍵1の運動をハンマー2に伝達するアクション3と、ハンマー2によって打弦される弦4と、鍵1を駆動するソレノイド5とを有している。そして、ソレノイド5のプランジャが突出すると、鍵1がバランスピンPを中心に回転し、その演奏者側が下がり（以下、この状態を押鍵状態という）、また、これに連動してアクション3が作動し、ダンパー6が弦4から離れるとともに、ハンマー2が回転して打弦する。一方、演奏者が弾く場合は、指で鍵1を押下することにより、上述と同様の作用が生じて打弦が行われる。

【0019】なお、図において、SE1、SE2は、打弦速度を計測するためのセンサであり、ハンマー2がこれらのセンサSE1、SE2の間を通過する時間を計測することにより、ハンマー2の速度、すなわち、打弦速度が計測される。また、ハンマー2がセンサSE1を通過する時刻が打弦時刻として計測される。ところで、鍵1を押下げる速度に応じてハンマー2の打弦速度が決まるが、鍵1の速度は初め遅くて次第に早くなる場合や、その逆の場合もあり、さらには、ほとんど一定の速さで押される場合もある。この場合、鍵1のレスト位置（鍵1を押していない場合の初期位置）からエンド位置（鍵を押した位置）に至るまでの速度と、ハンマー2の打弦速度とがどのような関係になっているのかが重要である。なぜならば、その関係を考察せず、打弦強度データに応じて鍵速度（初期速度など）を制御しても、記録時の打弦速度を再生することはできないからである。

【0020】実験によれば、鍵1のある位置における速度とハンマー2の打弦速度とが極めて良い対応を示すことが判った。この位置は、ピアノの個体差にもよるが、概ねレスト位置から9.0mm～9.5mm程度押し下げた位置であった。したがって、鍵1がこの位置に達するときの速度を、打弦強度データに応じて制御すれば、記録時の打弦速度を忠実に再現することができる。なお、以下においては、上述の所定位置をリファレンスポイントXrという。

【0021】②リファレンス速度

次に、上述のようにして求めたリファレンスポイントXrにおいて、どのような鍵速度にすれば、打弦速度を忠実に再現することができるかを設定する必要がある。なお、以下においては、リファレンスポイントXrにおける鍵速度をリファレンス速度Vrという。ここで、図2はリファレンスポイントXrを9.5mmに設定したときの鍵速度と打弦速度の関係を示す図である。図中、白点は鍵をエンド位置まで押し切る単打奏法を行った場合の結果を示し、黒点は鍵をエンド位置まで押し切らずに連打する連打奏法を行った場合の結果を示している。また、C1は1次最小自乗法近似による直線、C2は6次最小自乗法による曲線を示している。

【0022】図2から明らかなように、リファレンス速度Vrは、直線C1あるいは曲線C2のいずれによっても近似できる。したがって、近似性のよい関数を適宜選択すれば、この関数を用いて任意の打弦強度データ（記録時の打弦速度情報）からリファレンス速度Vrを決定することができる。この実施例においては、計算が簡単で誤差の少ない1次関数近似を採用している。したがって、リファレンス速度Vrは、次式によって求められる。

【0023】

$$\text{【数1】 } V_r = \alpha \cdot V_H + \beta$$

数1において、VHは打弦速度（打弦強度データ）であり、 α および β は定数である。定数 α および β は、ピアノの機種等に応じ実験等によって決定する。なお、 α および β は、同一ピアノであっても、リファレンスポイントXrをどこにするかによって変動する。

【0024】③リファレンス時間差

さて、演奏データに含まれる打弦時刻データは、前述したように、相対時刻あるいは絶対時刻で記録されているが、いずれにしても再生側自動ピアノにおいて打弦時刻データを読みとって処理することにより、再生時の各音の打弦絶対時刻が求められる。そこで、このようにして求めた打弦絶対時刻において正確に打弦を行わせるには、鍵が何時リファレンスポイントXrを通過すればよいかを求める必要がある。

【0025】ここで、鍵1がリファレンスポイントXrを通過する時刻（以下、リファレンス時刻trという）と打弦時刻（正確には、ハンマーが打弦位置直前にあるセンサSE1を通過した時刻）との時間差をリファレンス時間差Trと定義し、これと打弦速度との関係を実験により求めたものが図3である。図3において、白点は単打奏法による結果、黒点は連打奏法による結果を示している。そして、図3を縮尺2倍にしたものが図4であり、縮尺4倍にしたものが図5である。これらの図から判るように、リファレンス時間差Trと打弦速度との関係は、双曲線により極めて良好に近似される。すなわち、このリファレンス時間差Trは、打弦速度VHを分

母にする1変数式で近似することができ、次式によって算出される。

【0026】

【数2】 $T_r = -(\gamma/v_H) + \delta$

なお、数2における定数 γ および δ は、ピアノの機種等に応じ実験等によって決定する。なお、 γ および δ は、同一ピアノであっても、リファレンスポイント X_r をどこにするかによって変動する。これは、数1における α 、 β の場合と同様である。

【0027】さて、数2によって、リファレンス時間差 T_r が求まれば、再生側の打弦絶対時刻からリファレンス時間差 T_r を減算することによって、リファレンス時刻 t_r が求められ、結局、上述した①、②、③の処理により、リファレンスポイント X_r 、リファレンス速度 V_r 、およびリファレンス時刻 t_r が求められる。したがって、リファレンス時刻 t_r にリファレンスポイント X_r に達し、かつ、その時の速度がリファレンス速度 V_r となるように鍵1を駆動すれば、記録時の打弦状態を忠実に再現することができると判る。なお、鍵1がリファレンスポイント X_r に達したときに打弦が行われるのであれば、リファレンス時間差 T_r を求める処理は不要になる。

【0028】以上のようにして、予め設定したリファレンスポイント X_r におけるリファレンス速度 V_r が求められれば、鍵1の挙動をこれに応じて制御することにより、記録時の打弦速度を再生することができる。この場合、レスト位置からエンド位置までの鍵の軌道（時間経過に対する鍵の位置）を設定すれば、鍵の押下開始位置からリファレンスポイント X_r に至るまでの各位置と時刻の関係（速度と時刻の関係等）が求められるから、これに応じて鍵の位置をフィードバック制御すればよいことが判る。この場合における鍵の軌道は、例えば、リファレンスポイント X_r を通る直線軌道（等速運動の場合）や放物線軌道（等加速度運動の場合）の他、任意の軌道を設定することができる。ただし、打弦状態の再現性が良く、また、制御し易い軌道を選択することが好ましい。また、設定する軌道によっては、リファレンス点における加速度等（運動を決定する他の要素）を求める必要が生じるが、これについては軌道の種類に応じて適宜算出すればよい。

【0029】④ダミーイベントの追加

ところで、上述したような再生処理を実現するためには、演奏記録時において、押鍵時刻、打弦時刻および打弦速度を得ておく必要がある。しかし、演奏状態あるいは自動ピアノの性能等により、押鍵イベントあるいはハンマーイベント等が抜ける虞がある。その詳細を図21、図22を参照し説明する。まず、図21(a)は、演奏記録時における鍵1の押鍵深さを示している。図において一点鎖線 L_1 は、センサによって押鍵または離鍵イベントが検出される押鍵深さである。また、同図(b)

は、押鍵深さの軌道が一点鎖線 L_1 と交差するタイミング、すなわち押鍵または離鍵イベントが発生するタイミングを示している。また、同図(c)は、センサSE1、SE2によりハンマーイベントの検出された時刻（正確には、ハンマーが打弦位置直前にあるセンサSE1を通過した時刻）と、検出されたハンマー速度 V_{h1} 、 V_{h2} を示す。

【0030】図21(a)によれば、時刻 t_1 から鍵1の押鍵が開始され、時刻 t_2 において押鍵イベントが検出されている。また、時刻 t_3 においては、この押鍵動作に対応するハンマーイベントが生じている。次に、時刻 t_4 においては、鍵1が若干離鍵されているが、離鍵イベントが検出される前に再び押鍵が開始され、この押鍵動作に対応するハンマーイベントが時刻 t_5 に生じている。すなわち、ハーフストロークによる押鍵が行われている。そして、その後鍵1が離鍵され、時刻 t_6 において離鍵イベントが検出されている。すなわち、図21(a)～(c)によれば、時刻 t_1 のハンマーイベントに対応する離鍵イベントと、時刻 t_5 のハンマーイベントに対応する押鍵イベントは存在しない。このような状態を、以下「イベント抜け」という。

【0031】また、図22(a)～(c)においても、ハーフストローク時における押鍵深さ、鍵イベントおよびハンマーイベントが示されている。まず時刻 t_1 においては押鍵イベントが検出され、これに対応するハンマーイベントおよび離鍵イベントが時刻 t_2 、および時刻 t_3 に各々検出されている。次に、時刻 t_4 においては、ハーフストロークによる押鍵イベントが検出されているが、この押鍵は、ダンパー操作を行うための弱い押鍵であるため、ハンマー2は打弦を行う前にレスト位置に戻っている。従って、同図(c)を参照すると、時刻 t_4 の押鍵イベントに対応するハンマーイベントは存在しない。すなわち、ハンマーイベントのイベント抜けが生じている。

【0032】本実施例においては、以上のようなイベント抜けが検出された場合には、仮想のイベント（ダミーイベント）が適宜追加される。すなわち、第1のハンマーイベントと第2のハンマーイベントとの間において、離鍵イベントおよび押鍵イベントが抜けている場合には、第1のハンマーイベントの発生時刻において、ダミーの離鍵イベントとダミーの押鍵イベントとが追加される。例えば、図21(d)においては、ハンマーイベントの発生する時刻 t_1 においてダミーの離鍵イベントとダミーの押鍵イベントとが追加されている。一方、押鍵イベントと離鍵イベントとの間においてハンマーイベントが抜けている場合は、当該押鍵イベントの発生時刻において弱いハンマーイベントが追加される。ここで、「弱いハンマーイベント」とは、発音が行われない程度に、きわめてハンマー速度が遅いイベントをいう。例えば、図22(d)においては、押鍵イベントが発生する時

刻 t_r において、ダミーの弱いハンマーイベントが追加されている。ダミーイベントの追加は、押鍵イベント、ハンマーイベント、離鍵イベントの順序、あるいはイベントの時間間隔などに応じて行われる。その詳細については後述する。

【0033】以上のように追加されるダミーイベントの発生時刻は既存のイベントと同一の時刻であるが、これら発生時刻は適切なリファレンス軌道を生成するために、適宜変更される。その詳細については後述する。また、これらダミーイベントを追加する理由を説明しておく。まず、本実施例においては、上述したように、演奏記録時において、押鍵時刻、打弦時刻および打弦速度を得ておく必要がある。従って、これらのうち一部が抜けた場合には、正確なリファレンス軌道を生成することが困難になる。例えば、本実施例において、各種のダミーイベントを追加することにより、図21(e)、図22(e)、図23(f)の実線に示すようなリファレンス軌道を得ることができる(詳細は後述する)。これに対して、ダミーイベントを追加しなかった場合には、例えば同図の破線 $L_1 \sim L_4$ に示すような、実際の演奏状態と大幅に異なるリファレンス軌道が生成されることになる。すなわち、本実施例においては、ダミーイベントを適宜追加することにより、実際の演奏状態に近いリファレンス軌道が得られるのである。

【0034】⑤処理の概要

以上のようにして、予め設定したリファレンスポイント X_r におけるリファレンス速度 V_r が求められれば、鍵1の挙動をこれに応じて制御することにより、記録時の打弦速度を再生することができる。また、イベント抜けが検出された場合には適宜ダミーイベントが追加されるから、ハーフストローク等の奏法が採られた場合においても、安定して演奏状態を再生することが可能である。

【0035】この場合、レスト位置からエンド位置までの鍵の軌道(時間経過に対する鍵の位置)を設定すれば、鍵の押下開始位置からリファレンスポイント X_r に至るまでの各位置と時刻の関係(速度と時刻の関係等)が求められるから、これに応じて鍵の位置をフィードバック制御すればよいことが判る。この場合における鍵の軌道は、例えば、リファレンスポイント X_r を通る直線軌道(等速運動の場合)や放物線軌道(等加速度運動の場合)の他、任意の軌道を設定することができる。ただし、打弦状態の再現性が良く、また、制御し易い軌道を選択することが好ましい。また、設定する軌道によっては、リファレンス点における加速度等(運動を決定する他の要素)を求める必要が生じるが、これについては軌道の種類に応じて適宜算出すればよい。以上がこの実施例の制御原理である。

【0036】B. 実施例の構成

B-1. 全体構成

次に、本実施例の全体構成を説明する。図1は、この実

施例の構成を示すブロック図である。なお、前述した図20の各部と対応する部分には同一の符号を付け、その説明を省略する。

【0037】図において、10は再生前処理部であり、記録メディアあるいはリアルタイム通信装置から供給される演奏データに基づいて、鍵の軌道データを作成する回路であり、まず、リファレンス時刻 t_r 、リファレンス速度 v_r を求め、さらに、加速度の推定が必要な場合はこれを演算する。そして、軌道の種類(直線、放物線、その他任意の軌道)に応じて各音に対する鍵軌道を生成する。11は、再生前処理部10が作成した軌道データに基づいて、ソレノイド5のブランジャの動きを示すデータを作成するモーションコントローラである。また、12はサーボコントローラであり、モーションコントローラ11から供給されるデータに応じてソレノイド5の励磁電流を制御する。この場合、ソレノイド5には、ブランジャの位置を検出する検出機構が設けられており、これにより検出されたブランジャ位置をフィードバック信号としてサーボコントローラ12にフィードバックするようになっている。そして、サーボコントローラ12は、モーションコントローラ11から供給されるデータとソレノイド5からのフィードバック信号とを照合しながら、両者が一致するように励磁電流を制御する。

【0038】次に、26は、鍵1の下面に取り付けられた板状のシャッタである。25は、上下方向に所定距離隔て設けられる2組のフォトセンサによって構成されているキーセンサであり、鍵1が押下され始めると、まず上方のフォトセンサが遮光され、次いで、下方のフォトセンサが遮光される。離鍵の際には、下方のフォトセンサが受光状態になり、ついで、上方のセンサが受光状態になる。

【0039】キーセンサ25の出力信号は、演奏記録部30に供給され、演奏記録部30は、キーセンサ25内の下方のフォトセンサが受光状態になってから上方のフォトセンサが受光状態になるまでの時間を測定し、ここから、離鍵速度を検出する。また、演奏記録部30は上方のフォトセンサが受光状態になった時刻を離鍵時刻として処理する。また、演奏記録部30は、センサSE2が遮光されてからセンサSE1が遮光されるまでの時間を計測し、ここから打弦速度を検出する。また、センサSE1が遮光された時刻を打弦時刻として処理する。

【0040】すなわち、演奏記録部30は、演奏が開始されると、センサSE1、SE2の出力信号に基づいて、打弦時刻および打弦速度を検出し、かつ、キーセンサ25の出力信号に基づいて離鍵時刻および離鍵速度を検出する。以上のようにして検出された各情報は、記録後処理部31に供給される。

【0041】記録後処理部31においては、演奏記録部30から供給される各種情報に対し、正規化処理を施し

た後に、外部の記録媒体に演奏データとして供給する。ここで、正規化処理とは、ピアノの個体差を吸収するための処理である。すなわち、打弦速度、打弦時刻、離鍵速度、離鍵時刻等は、各ピアノにおけるセンサの位置や、構造上の違い、あるいは機械的誤差によって固有の傾向を持つため、標準となるピアノを想定し、そのピアノにおける打弦速度、打弦時刻等に変換するための処理である。また、記録後処理部31は、演奏記録部30で記録された情報にイベント抜けを検出すると、演奏データに適宜ダミーイベントを追加する（詳細は後述する）。

【0042】上述した構成において自動ピアノを演奏すると、演奏記録部30においてハンマーイベントおよび鍵イベント等が記録され、記録後処理部31においてこれらの情報に正規化処理等が施された後、演奏データが出力される。この演奏データは、他の自動楽器に供給され、あるいは適当な記録媒体に記録される。次に、記録された演奏データを再生すると、演奏データ中の打弦強度データに応じて、リファレンスポイントX_rを通過する鍵軌道が生成され、鍵がこの軌道を再現するようにソレノイド5の励磁電流が制御される。この結果、鍵1がリファレンスポイントX_rに達したときの速度は、記録時の打弦速度に対応した速度（すなわち、リファレンス速度V_r）になり、ハンマー2は記録時と同じ速度で打弦を行う。

【0043】以上のように、この実施例においては、リファレンスポイントという制御点を導入し、この点における鍵の運動属性を再現するとともに、イベント抜けが生じた場合は演奏データに適宜ダミーイベントを追加するため、センサ等を増やすことなく、忠実に打弦速度を再現することができる。

【0044】C. 実施例の動作

C-1. 演奏記録動作

演奏記録を行う場合、演奏者は、まず演奏記録部30を動作状態にしておき、自動ピアノを演奏する。これにより、演奏記録部30においてハンマーイベントおよび鍵イベント等が記録される。次に、演奏者が記録後処理部31を動作させると、図6に示すプログラムが実行される。図において処理が開始されると、ステップSP100において各種の変数の初期設定が行われる。次に処理がステップSP101に進むと、演奏記録部30からハンマーイベントおよび鍵イベント等のデータが読出される。また、ステップSP101においては、読出されたデータに対して対応関係にあるデータも併せて読出される。例えば、最初に読出されたデータが所定のキーコードKCの押鍵イベント（押鍵時刻）であった場合、そのキーコードKCの離鍵イベント（離鍵時刻）およびハンマーイベント（打弦時刻、打弦速度）が検索され、存在するならばこれらのデータも演奏記録部30から読出される。

【0045】①一般的な演奏状態に対する動作

次に、処理がステップSP102を介してステップSP103に進むと、先に読出されたデータの対応関係がどのような状態であるか判定される。一般的な（正常な）演奏状態では、押鍵イベントと離鍵イベントとの間に1回のハンマーイベントが存在する。かかる場合は処理がステップSP104に進む。ステップSP104においては、押鍵時刻が打弦時刻以下であるか否かが判定される。一般的な状態では押鍵イベントが生じた後にハンマーイベントが生じるから「YES」と判定され、処理がステップSP105に進む。ステップSP105においては、押鍵時刻および打弦時刻の時間差が所定値A以上であるか否かが判定される。

【0046】ここで、所定値Aは一般的な押鍵状態における押鍵時刻、打弦時刻間の時間よりも若干余裕を持った値、例えば「10 msec」に設定されている。従って、一般的な演奏状態では「NO」と判定され、処理がステップSP113に進む。ステップSP113においては、先にステップSP101で読出されたデータに上述した正規化処理が施される。すなわち、ステップSP101で読出された打弦速度、打弦時刻、離鍵速度、離鍵時刻等のデータが、標準となるピアノにおける打弦速度、打弦時刻等に変換されるのである。そして、このデータは、ステップSP114において、演奏データとして外部に出力される。出力された演奏データは磁気ディスク等の記録媒体に適宜記録される。

【0047】以上の処理が終了すると、処理がステップSP101に戻り、演奏記録部30から次のデータが読出され、上述したのと同様の処理が行われる。そして、演奏記録部30に記録されたデータの全てが既に読み出され、ステップSP101において新たなデータを読み出せなかった場合は、ステップSP102で「YES」と判定され、記録後処理部31の動作が終了する。このように、一般的な演奏状態においては、演奏記録部30から読出されたデータに対して単に正規化処理が施され、その結果が演奏データとして出力されることが判る。

【0048】②遅い押鍵に対する動作

演奏記録時において何れかの鍵がゆっくりと押下されると、押鍵時刻から打弦時刻までの時間が所定値Aよりも長くなる場合がある。かかる演奏状態が演奏記録部30に記録された場合は、ステップSP105において「YES」と判定され、処理はステップSP106に進む。ステップSP106においては、所定値Bが押鍵速度で除算され、その商は押鍵時刻、打弦時刻の時間差以上であるか否かが判定される。ここで「NO」と判定されると、処理はステップSP113に進む。すなわち、押鍵速度が遅い場合には、時間差が所定値Aを超えている場合であっても、この時間差が押鍵速度に鑑みて妥当な値であれば、一般的な場合と同様に処理をステップSP1

13に進めることとしたものである。ここで、所定値Bは、例えば「10mm」（所定値Bのディメンションは $s \cdot mm / s = mm$ になる）程度に設定するとよい。

【0049】③離鍵／押鍵イベント抜けに対する動作

先に図21について説明したように、ハーフストロークの連打を行った場合等においては、複数のハンマーイベントの間に離鍵／押鍵イベントが存在しない場合がある。かかる状態のデータが読出され処理がステップSP103に進むと、離鍵／押鍵イベント抜けであると判定され、処理がステップSP111に進む。ステップSP111においては、最後のハンマーイベントを除く全てのハンマーイベントに対応して、各打弦時刻と同一時刻のダミーの離鍵および押鍵イベントが追加される。

【0050】次に、処理がステップSP112に進むと、追加されたダミーの離鍵および押鍵イベントの内容が修正される。その内容を、以下詳述する。まず、離鍵／押鍵イベント抜けが生じた場合においても、ハンマーイベントの内容、すなわち打弦時刻および打弦速度は既知である。また、先に図2～図5において説明したように、打弦速度が既知であればリファレンス速度 V_r およびリファレンス時間差 T_r を求めることができる。従って、「打弦時刻よりもリファレンス時間差 T_r だけ前の時刻にリファレンス点を通過し、かつ、リファレンス速度 V_r の傾きを有する直線」は一意に決定される。

【0051】本実施例においては、押鍵時の鍵軌道は上記直線に沿った等速打鍵のものと仮定し、これによってダミーの押鍵イベントの内容が修正される。すなわち、ダミーの押鍵イベントについて、押鍵時刻は上記直線がセンサ位置と交差する時刻に設定され、押鍵速度はリファレンス速度 V_r に等しくなるように（すなわち等速打鍵になるように）設定される。

【0052】また、ダミーの離鍵速度は、当該自動ピアノの機構上実現できる最大の離鍵速度、すなわち演奏者が鍵に全く触れていない場合の離鍵速度に設定される。この場合、ハンマーイベントが生じた後に直ちに離鍵が開始されたものと仮定する。従って、離鍵時刻は、ハンマーイベントを起点とする最大傾斜の離鍵軌道がセンサ位置と交差する時刻に設定される。このように、ダミーの離鍵および押鍵イベントが追加され、これらの内容が修正されると、処理がステップSP113に進み、一般的な場合と同様の処理が行われる。

【0053】④ハンマーイベント抜けに対する動作

先に図22の時刻 $t_1 \sim t_2$ について説明したように、ハーフストローク奏法を行った場合は、押鍵イベントおよび離鍵イベントは生ずるが、これらに対応するハンマーイベントは発生しないことがある。かかる状態のデータが読出され処理がステップSP103に進むと、ハンマーイベント抜けであると判定され、処理がステップSP107に進む。ステップSP107においては、押鍵時刻と同一時刻にダミーのハンマーイベントが追加され

る。

【0054】次に、処理がステップSP108に進むと、ダミーのハンマーイベントに係る打弦速度が微少値、すなわち発音されない程度の値に設定される。また、かかる微少な打弦速度に対応してリファレンス速度 V_r およびリファレンス時間差 T_r が求まるから、上記ステップSP112と同様に、「打弦時刻よりもリファレンス時間差 T_r だけ前の時刻にリファレンス点を通過し、かつ、リファレンス速度 V_r の傾きを有する直線」が一意に決定され、その内容に基づいて押鍵イベントの内容が修正される。このように、弱いハンマーイベントが追加されるとともに押鍵イベントの内容が修正されると、処理がステップSP113に進み、一般的な場合と同様の処理が行われる。

【0055】⑤押鍵時刻、打弦時刻の時間差が大き過ぎる場合の処理

押鍵時刻、打弦時刻の時間差が大き過ぎる場合の具体例を、図23(a)～(c)に示す。これらの図において、鍵軌道は時刻 t_1 にセンサ位置に達しており、この時刻 t_1 に押鍵イベントが発生している。しかし、この時の押鍵動作はダンパー操作を行うためのハーフストロークであり、ハンマーイベントは発生していない。その後、センサ位置近くまで鍵が戻されるが、センサ位置に達する前の時刻 t_2 において再び鍵が押下され、時刻 t_3 にハンマーイベントが発生し、時刻 t_4 に離鍵イベントが発生している。

【0056】上記例では、押鍵イベントと離鍵イベントの間に1回のハンマーイベントが存在する。従って、ステップSP103においては、各イベント間の関係は一応正常なものと判定され、処理はステップSP104に進む。しかし、上記例では押鍵時刻と打弦時刻の時間差が大き過ぎるため、ステップSP105、106において何れも「YES」と判定され、処理がステップSP107に進む。ステップSP107においては、上記「ハンマーイベント抜け」の場合と同様に、ダミーの弱いハンマーイベントが追加される（同図(d)参照）。しかし、ダミーのハンマーイベントを追加したために、押鍵イベントと離鍵イベントの間に2回のハンマーイベントが生ずることになる。これは、「離鍵／押鍵イベント抜け」の場合と同様の状態であるため、ダミーのハンマーイベントと同一時刻に、ダミーの離鍵および押鍵イベントがさらに追加される（同図(e)参照）。

【0057】次に、処理がステップSP108に進むと、ダミーのハンマーイベントに係る打弦速度が微少値に設定され、これに伴って時刻 t_1 に発生した押鍵イベントの内容も修正される。そして、以上の処理が終了すると、処理がステップSP113に進み、一般的な場合と同様の処理が行われる。また、ダミーの離鍵および押鍵イベントは、前述した離鍵／押鍵イベント抜けに対する動作と同様にその内容が修正される。

【0058】⑥打弦時刻が押鍵時刻よりも早い場合の処理

一般的な演奏状態では、最初に押鍵イベントが生じた後、ハンマーイベントが発生する。従って、打弦時刻が押鍵時刻よりも早くなることは、ほとんど起り得ないと考えられる。しかし、例えば同一の鍵を高速に連打した場合などにおいて、ハンマー2やアクション3の運動状態によっては、かかる事態が生じる可能性も否定できない。

【0059】かかる場合は、ステップSP104において「NO」と判定され、処理がステップSP109に進む。ステップSP109においては、打弦時刻と押鍵時刻の時間差が所定値C（例えば10msec）以上であるか否かが判定される。ここで「YES」と判定されると、ステップSP110において所定値D（例えば10mm）が打弦速度で除算され、その商は上記時間差以上であるか否かが判定される。

【0060】ここで、打弦時刻と押鍵時刻との時間差が大きい場合には、ステップSP109、110の何れにおいても「YES」と判定されるのであるが、かかる場合にはハンマーイベントは押鍵イベントから独立したものとして推定することができる。すなわち、同一鍵の高速連打等を行った場合はハンマー2やアクション3が特殊な運動状態になるため、レスト位置からセンサ位置直前までの短い距離の押鍵動作（この押鍵動作はキーセンサ25によっては検出されない）によってもハンマーイベントが生じることも考えられる。かかる打弦動作が行われた後、ダンパー操作のみを行う弱い押鍵動作（この操作によって押鍵／離鍵イベントは生じるが、ハンマーイベントは生じない）が行われると、結果的にハンマーイベント、押鍵イベント、および離鍵イベントの順で各イベントが発生する。

【0061】そこで、かかる場合は、処理をステップSP111に進め、ハンマーイベントと同時にダミーの押鍵／離鍵イベントを追加するとともに、その後に生じる押鍵イベントと離鍵イベントとの間にダミーの弱いハンマーイベントを追加することとしている。

【0062】一方、打弦時刻と押鍵時刻との時間差が小さい場合には、ステップSP109、110の何れかにおいて「NO」と判定されるのであるが、かかる場合には押鍵イベントとハンマーイベントとがなんらかの関係を有していると推定されるため、各々独立にダミーイベントを追加したのでは、かえって不都合な結果を招くことも考えられる。そこで、かかる場合は、処理をステップSP113に進め、ステップSP101において読出されたデータに対して、ダミーイベントを追加することなく正規化処理を施すこととしている。なお、ハンマーイベントの後に押鍵イベントが発生するような状態では、後にリファレンス軌道を生成する際に不都合が生じるため、押鍵時刻が打弦時刻の前になるように押鍵イベ

ントの内容を変更しておくことと良い。そして、処理がステップSP112に進むと、押鍵イベントおよび離鍵イベントの内容が修正される。

【0063】なお、上述した各処理においては、ダミーのイベントを適宜追加するために、現実にはあり得ない順番で鍵イベントが配列されることがある。例えば、図22(e)に示す例にあっては、時刻 t_{11} に押鍵イベントが発生した後、再び時刻 t_{12} に押鍵イベントが発生し、時刻 t_{13} 、 t_{14} においても連続して離鍵イベントが発生している。従って、どの押鍵イベントがどの離鍵イベントと対応するのかを明確にするために、演奏データを外部に出力する際に識別情報を付与しておくこととよい。また、同図に示すように、複数の押鍵／離鍵イベントに基づいて生成されるリファレンス軌道は相互に交差することもあるが、その対策については後述する。

【0064】C-2. 演奏再生動作

さて、上述のように記録後処理部31を介して出力された演奏データは磁気ディスク等の記録媒体に適宜記録されるが、この演奏データに基づいて自動演奏を行う場合には、演奏データを再生前処理部10に供給する。以下、本実施例における再生前処理部10について説明する。この実施例においては、鍵軌道として直線を想定するので、次のような手法で軌道データの作成を行う。

【0065】①押鍵時の軌道データ作成

図7は、鍵の押鍵軌道（直線軌道）を示す図であり、レスト位置 X_0 から等速運動をしてエンド位置 X_e に至っている。ここで、鍵の初速度を V_0 、鍵の位置を X 、鍵の駆動開始時点からの時刻を t とすれば、鍵の軌道は、

【0066】

$$\text{【数3】 } X = V_0 \cdot t + X_0$$

と表される。また、鍵がリファレンスポイント X_r に達する時刻を t_r とすると、

【0067】

$$\text{【数4】 } X_r = V_0 \cdot t_r + X_0$$

なる式が成り立つから、この数4から時刻 t_r を求めることができる。したがって、押鍵を開始する絶対時刻（以下、押鍵開始時刻という） t_0 は、次式によって求めることができる。

【0068】

$$\text{【数5】 } t_0 = t_r - t_r'$$

$$= t_r - (X_r - X_0) / V_0$$

なお、リファレンス時刻 t_r は、前述のように、打弦時刻からリファレンス時間差 T_r を減算することによって求める。上記数5によって押鍵開始時刻 t_0 を求め、この時刻から、数3で示される軌道に従って鍵1を駆動すれば、鍵1は、リファレンス時刻 t_r において正確にリファレンスポイント X_r に達し、しかも、その時の速度は、打弦強度データに対応したリファレンス速度 V_r となる。

【0069】なお、鍵の挙動については、直線軌道（等

速運動)を想定しているから、リファレンス速度 V_r と初速度 V_0 は等しい。そして、リファレンス速度 V_r は、前述の数1によって求められるから、数5で求めた押鍵開始時刻 t_0 から一定速度 V_r で鍵を駆動するように制御(速度制御)しても上記と同様の結果を得ることができる。

【0070】②離鍵時の軌道データ作成

次に、離鍵時の軌道データ作成について説明する。まず、鍵の位置を X_N 、離鍵初速度を $V_0N(<0)$ 、離鍵開始時点からの時刻を t_N 、エンド位置を X_e とすれば、離鍵時の鍵軌道は、次式で表される。

【0071】

$$【数6】X_N = V_0N \cdot t_N + X_e$$

ここで、図8は数6で示される軌道を示す図である。さて、前述のように、演奏記録部30(図1参照)は、キーセンサ25内の下方のフォトセンサが受光状態になってから上方のフォトセンサが受光状態になるまでの時間を測定して離鍵速度 V_{kN} を検出し、また、上方のフォトセンサが受光状態になった時刻を離鍵時刻 t_{kN} として検出する。この場合、離鍵時刻 t_{kN} におけるダンパー6は、弦4に接して音の減衰を開始する状態になっている(そのような状態になるようフォトセンサの位置が調整されている)。そして、このようにして検出された離鍵速度 V_{kN} および離鍵時刻 t_{kN} は、それぞれ演奏情報を構成するデータとして記録され、再生時に読み出される。

【0072】ここで、ダンパー6が弦4に接するときの鍵の位置を離鍵リファレンスポイント X_{rN} と定義すれば、鍵1が離鍵リファレンスポイント X_{rN} に達したときに、離鍵状態になったとすることができる。したがって、鍵1が離鍵リファレンスポイント X_{rN} に達する時刻(以下、離鍵リファレンス時刻 t_{rN} という)と、演奏情報中の離鍵時刻 t_{kN} とが一致するように鍵位置を制御すれば、正確な離鍵タイミング制御を行うことができる。

【0073】また、ダンパー6が弦4に接する速さは、音の減衰状態に影響を与えるから、これを忠実に再現することが望ましい。この速さは、離鍵速度 V_{kN} に対応するから、結局、離鍵リファレンスポイント X_{rN} における鍵速度(以下、離鍵リファレンス速度 V_{rN} という)を正確に離鍵速度 V_{kN} に一致させれば、音の減衰状態が正確に再現される。ここで、鍵の駆動が開始される時刻を基準($=0$)にして、鍵がリファレンスポイント X_{rN} に達する時刻を t_{rN}' とすると、

【0074】

$$【数7】X_{rN} = V_0N \cdot t_{rN}' + X_eN$$

(ただし、直線軌道だから $V_0N = V_{rN} = V_{kN}$)

なる関係が成り立ち、この数7より時刻 t_{rN}' を求めることができる。したがって、次式によって離鍵開始時刻 t_0N を求めることができる。

【0075】

$$【数8】t_0N = t_{rN} - t_{rN}'$$

$$= t_{rN} - (X_{rN} - X_eN) / V_0N$$

この数8によって離鍵開始時刻 t_0N を求め、この時刻から、数6で示される軌道に従って鍵を駆動すれば、鍵は離鍵時刻 t_{kN} において離鍵リファレンスポイント X_{rN} に達し、記録時の離鍵状態を忠実に再現することができる。なお、時刻 t_0 から速度 $V_0N (= v_{kN}$:離鍵速度)で鍵駆動するように制御(速度制御)しても上記と同様の結果を得ることができる。

【0076】③交差時の軌道データ作成(交差処理)

押鍵軌道および離鍵軌道は上述のようにして作成されるが、鍵の操作には、離鍵の途中から次の押鍵に移ったり、あるいは、押鍵の途中から離鍵され場合がある。このような場合においては、作成した押鍵軌道と離鍵軌道とが交差する。例えば、図9はこのような軌道の交差状態を示しており、図示の状態では、時刻 t_1 から時刻 t_c まで押鍵が行われ、時刻 t_1 から時刻 t_c まで離鍵が行われている。このとき、上述の方法によって生成される押鍵軌道は、時刻 t_1 にレスト位置 X_0 を離れ、時刻 t_c においてエンド位置 X_e に達する軌道であり、また、離鍵軌道は時刻 t_1 にエンド位置 X_e を離れ、時刻 t_c においてレスト位置 X_0 に達する軌道である。

【0077】ここで、交差する時刻 t_c を求めることができれば、 $t_1 \sim t_c$ までは押鍵軌道に基づいて鍵1を制御し、 $t_c \sim t_c$ までは離鍵軌道に基づいて鍵を制御すればよい。ここで、図9に示すような、押鍵の後に発生した離鍵の軌道が交差する場合は、交差時刻 t_d は次のようにして求めることができる。

【0078】

【数9】

$$t_c = (V_0 \cdot t_1 - V_0N \cdot t_0N) / (V_0 - V_0N)$$

$$= t_0N + V_0 \cdot (t_1 - t_0N) / (V_0 - V_0N)$$

なお、数9における t_1 は、次式により算出される。

【0079】

$$【数10】t_1 = t_0 + (X_e - X_0) / V_0$$

また、離鍵の後に発生した押鍵の軌道が交差する場合も、2つの直線軌道の交点を求めればよいので、上記と同様の考え方により交差時刻を求めることができる。また、上述したように、本実施例にあっては適宜ダミーイベントを追加するから、図22(e)のように、押鍵軌道同士あるいは離鍵軌道同士が交差することも考えられる。押鍵軌道同士が交差する場合には、同図に示すように、先にエンド位置に達する軌道(押鍵速度の速い軌道)を採用するのが妥当である。これは、強いアタックを忠実に再現するためである。一方、離鍵軌道同士が交差する場合には、最初に離鍵速度の速い軌道を採用し、交差点を過ぎた後は、離鍵速度の遅い軌道を採用するとよい。

【0080】このようにして、交差時刻を求め、押鍵軌

道と離鍵軌道を組み合わせることにより、交差時の軌道データを作成する。以上が、再生前処理部10における軌道データの作成であり、このようにして作成された軌道データは、図1に示すモーションコントローラ11に供給される。モーションコントローラ11においては、作成された軌道データに基づいて、各時刻における鍵1の位置に対応した位置制御データ(X)を作成し、サーボコントローラ12に供給する。サーボコントローラ12は、位置制御データ(X)に応じた励磁電流をソレノイド5に供給するとともに、ソレノイド5から供給されるフィードバック信号と制御データ(X)を比較し、両者が一致するようにサーボ制御を行う。

【0081】④作成された軌道データ的具体例

以下、図21、図22、図23について、上述した軌道データを作成した場合の具体例を説明する。まず、図21(e)によれば、時刻 t_{11} 以前の再生リファレンス軌道は、レスト位置から等速運動をしてエンド位置に至っている。その速度はハンマー速度 V_h に対応したリファレンス速度である。その後、リファレンス軌道は、直ちに最大離鍵速度でレスト位置に向かっている。これは、ダミーの離鍵イベントが、当初においてはハンマーイベントと同一の時刻(時刻 t_{11} 、同図(d)参照)に追加されたためである。この離鍵時の軌道は、時刻 t_{11} のハンマーイベントに対応する押鍵軌道と交差しているため、上述した交差処理が行われる。この結果、リファレンス軌道は同図の実線に示すようになる。

【0082】次に、図22(e)によれば、時刻 t_1 の押鍵イベント、時刻 t_2 の離鍵イベントおよび時刻 t_{11} のハンマーイベントに対応して、時刻 t_2 および t_{11} に一点鎖線 L_1 と交差する軌道が生成されている。一方、同図(d)によれば、時刻 t_1 に弱いハンマーイベントが追加されているから、これに対応して傾きの小さな軌道(時刻 t_{11} に一点鎖線 L_1 と交差する軌道)が生成されている。両者は交差するから、上述した場合と同様に交差処理が行われる。

【0083】次に、図23(f)によれば、時刻 t_1 に至るまで、傾きの小さな押鍵軌道が生成されている。これは、時刻 t_1 に追加された弱いハンマーイベント(同図(d)参照)に対応するものである。そして、時刻 t_1 以降は、最大離鍵速度でレスト位置に向かう離鍵軌道が生成されている。その後、時刻 t_2 のハンマーイベントに対応して押鍵軌道が生成されており、その傾きはハンマー速度 V_h に対応したリファレンス速度に等しい。この例においては、押鍵/離鍵軌道は交差しないから、得られた押鍵/離鍵軌道がそのままリファレンス軌道になる。

【0084】⑤演奏再生の全体動作

次に、演奏再生時における実施例の全体動作を説明する。まず、再生前処理部10は、記録媒体から演奏情報を読み出し、その中の打弦時刻データおよび打弦速度デ

ータに基づいて、押鍵軌道を作成する。作成された押鍵軌道データは、モーションコントローラ11に供給され、ここで、位置制御データ(X)に変換される。すなわち、押鍵開始時刻 t_1 になると、数3に示される軌道データがモーションコントローラ11に供給され、モーションコントローラ11は、数3のXを時間経過とともに順次演算し、位置制御データ(X)を作成する。この位置制御データ(X)は、サーボコントローラ12に供給され、これにより、数3のXに対応した励磁電流がソレノイド5に供給される。この場合、サーボコントローラ12は、ソレノイド5のフィードバック信号と位置制御データ(X)とを比較し、両者が一致するように励磁電流を制御するから、ソレノイド5のブランジャの突出量は、数3のXに対応したものとなる。したがって、鍵1は数3で示される直線軌道に従って押下されていき、リファレンスポイント X_r においてリファレンス速度 V_r を有する運動を行う。これにより、記録時の打弦強度を忠実に再現する打弦が行われる。

【0085】次に、再生前処理部10は、演奏情報の中の離鍵時刻データ、離鍵速度データに応じて、離鍵軌道を作成し、モーションコントローラ11に供給する。モーションコントローラ11は、上述の場合と同様に、数6のX_Nを時間の経過に応じて順次演算し、位置制御データ(X)を作成し、サーボコントローラ12に供給する。この結果、サーボコントローラ12は数6に対応した励磁電流制御を行い、鍵1は数6で示される直線軌道に従ってレスト位置 X_r に戻っていく。これにより、鍵1の運動は、離鍵リファレンスポイント X_{rN} において離鍵リファレンス速度 V_{rN} を有するものとなり、記録時の離鍵状態が忠実に再現される。

【0086】一方、連打等が行われ、押鍵軌道と離鍵軌道が交差するときは、再生前処理部10が数10によって交差時刻 t_c を演算し、この時刻を境に、押鍵軌道と離鍵軌道を切り換えてモーションコントローラ11に供給する。これにより、鍵1は押鍵軌道の途中から離鍵軌道に、あるいは離鍵軌道の途中から押鍵軌道に切り換えられ、いわゆる、ハーフストロークの奏法に従った運動を行う。

【0087】II. 第2実施例

A. 実施例の制御原理

次に、この発明の第2実施例について説明する。この実施例が前述した第1実施例と異なる点は、再生前処理部10において放物線の軌道データを作成する点である。そこで、説明の簡略化のため、第2実施例については、再生前処理部10の動作原理および動作についてのみ説明する。この実施例における再生前処理部10は、押鍵時の鍵軌道として放物線、離鍵の鍵軌道として直線を想定するので、次のような手法で軌道データの作成を行う。

【0088】①押鍵時の軌道データ作成

押鍵軌道を次のように規定する。

【0089】

【数11】 $X = a/2 \cdot t^2 + b \cdot t + c$

ここで、 a は加速度であり、次式によって求める。

【0090】

【数12】

$a = (V_r - V_k) / (t_r - (t_{k_1} + t_{k_2}) / 2)$
数12におけるリファレンス速度 V_r は、前述の実施例と同様に、数1を用いて打弦速度から求める。また、 t_{k_1} は、次式によって求めることができる。

【0091】

【数13】 $t_{k_1} = t_{k_2} - X_d / V_k$

ここで、 X_d は、キーセンサ25内の2つのセンサの取り付け間隔である。また、図10は、この実施例における放物線軌道を示す図であり、図において、 X_{k_1} 、 X_{k_2} はキーセンサ25内の2つのセンサの取り付け位置である。この図からも判るように、キーオン速度 V_k は、シャッタ26が X_{k_1} と X_{k_2} の間を通過する間の平均速度であり、放物線軌道においては、中間時刻である時刻 $(t_{k_1} + t_{k_2}) / 2$ における速度である。前述した数12は、リファレンス速度 V_r とキーオン速度 V_k の速度差を、リファレンス時刻 t_r と中間時刻 $(t_{k_1} + t_{k_2}) / 2$ の時間差で除したもので、言い替えれば、リファレンスポイント X_r とセンサ中間点との間の速度変化から加速度を算出するものである。なお、加速度一定という推定のもとでは、任意の2点間の速度変化を基にして加速度を検出することができる。

【0092】以上のことから判るように、演奏データ中の打弦時刻と打弦速度から、リファレンス時刻 t_r とリファレンス速度 V_r が求められ、さらに、キーオン速度 V_k とキーオン時刻 t_{k_2} を参照して加速度 a が求められる。したがって、あとは押鍵開始時刻と初速度が求められれば、図10に示す放物線軌道に沿ってキーを駆動できることになる。次に、これらの求め方について説明する。まず、図11に示すように、押鍵開始時刻を t_0 、初速度を V_0 とすれば、数11における係数 b は、

【0093】

【数14】 $b = V_0 - a \cdot t_0$

と表され、また、数11における定数 c は、

【0094】

【数15】 $c = X_0 - (a/2) \cdot t_0^2 + b \cdot t_0$

と表される（ただし、数15における X_0 は前述のようにレスト位置）。ここで、押鍵開始時刻を基準（=0）にした場合のリファレンス時刻を $t_{r'}$ とし、さらに、数13、数14を用いて、数11をリファレンスポイント X_r における式に書き直すと次式になる。

【0095】

【数16】

$X_r = (a/2) \cdot t_{r'}^2 + V_0 \cdot t_{r'} + X_0$

ここで、リファレンス速度 V_r は、

【0096】

【数17】 $V_r = a \cdot t_{r'} + V_0$

と表すことができるから、この関係を数16に代入して整理すると、

【0097】

【数18】 $0 = (a/2) \cdot t_{r'}^2 - V_r \cdot t_{r'} - (X_0 - X_r)$

となり、ここから、 $t_{r'}$ を求めると、

【0098】

10 【数19】 $t_{r'} = (V_r - (V_r^2 + 2a(X_0 - X_r))^{1/2}) / a$

となる。したがって、加速度 a 、リファレンス速度 V_r 、リファレンスポイント X_r 、およびレスト位置 X_0 が既知であれば、鍵が押鍵開始時刻からリファレンスポイント X_r に至るまでの時刻 $t_{r'}$ が求められる。また、リファレンス時刻 t_r と押鍵開始時刻 t_0 の関係は、

【0099】

【数20】 $t_{r'} = t_r - t_0$

であるから、数20を用いて、押鍵開始時刻 t_0 が求められる。また、数16に数17を代入して初速度 V_0 について整理して解けば、

【0100】

【数21】 $V_0 = (V_r^2 + 2a(X_0 - X_r))^{1/2}$

となり、これにより初速度 V_0 を求めることができる。したがって、以上のようにして求めた初速度 V_0 、加速度 a を、数11に代入して軌道を求め、押鍵開始時刻 t_0 からその軌道に従って鍵を駆動することにより、打弦時刻と打弦速度とを正確に再現することができる。しかも、人間の鍵操作に近い放物線軌道なので、演奏者の微妙なニュアンスまで表現することが可能になる。なお、時刻 t_0 から初速度 V_0 で鍵を駆動し、後は、加速度 a に対応した速度変化分を用いて速度制御を行っても、上記と同様の効果が奏することができる。

【0101】②離鍵時の軌道データ作成

離鍵については、直線軌道を想定するので、前述した第1実施例と同様の離鍵軌道データ作成を行う。

【0102】③交差時の軌道データ作成

この実施例における押鍵軌道と離鍵軌道の交差は、放物線軌道と直線軌道との交差であるので、交差時刻 t_c は、次のようにして求めることができる。

【0103】

【数22】 $t_c = (-\Delta b + (\Delta b^2 - 2 \cdot \Delta a \cdot \Delta c)^{1/2}) / \Delta a$

ただし、数22における Δa は、

【0104】

【数23】 $\Delta a = a - a_N$

であり、 a_N は、離鍵時の加速度である。この実施例の場合は、離鍵時の加速度は0であるので、 $\Delta a = a$ （押鍵時の加速度）となる。また、数22における Δb は、

50 【0105】

【数24】

$$\Delta b = V_r - V_{rN} - (a \cdot t_r - a_N \cdot t_{rN}) \\ = V_r - V_{rN} - a \cdot t_r \quad (\because a_N = 0)$$

であり、 Δc は、

【0106】

【数25】 $\Delta c = X_r - X_{rN}$

$$+ (a \cdot t_r^2 - a_N \cdot t_{rN}^2) / 2 \\ - (V_r \cdot t_r - V_{rN} \cdot t_{rN}) \\ = X_r - X_{rN} \\ + (a \cdot t_r^2) / 2 \\ - (V_r \cdot t_r - V_{rN} \cdot t_{rN}) \quad (\because a_N = 0)$$

である。このようにして、交差時刻 t_c を求め、押鍵軌道と離鍵軌道を組み合わせることにより、交差時の軌道データを作成する。

【0107】以上が、第2実施例における再生前処理部10の軌道データ作成処理である。このようにして作成された軌道データは、図1に示すモーションコントローラ11に供給され、モーションコントローラ11においては、作成された軌道データに基づいて、各時刻における鍵1の位置に対応した位置制御データ(X)を作成し、サーボコントローラ12に供給する。このモーションコントローラ11とサーボコントローラ12の構成は、第1実施例と同様になっている。

【0108】B. 第2実施例の動作

次に、第2実施例の動作について説明するが、第2実施例の記録動作は第1実施例と同様である。また、再生動作も概ね第1実施例の動作と同じであるため、異なる点だけを説明する。

【0109】まず、再生前処理部10は、記録媒体から演奏情報を読み出し、その中の打弦時刻データ、打弦速度データ、キーオン速度データおよびキーオン時刻データに基づいて、放物線の押鍵軌道を作成する。作成された放物線の押鍵軌道データは、モーションコントローラ11に供給され、ここで、位置制御データ(X)に変換される。すなわち、押鍵開始時刻 t_0 になると、数11に示される軌道データがモーションコントローラ11に供給され、モーションコントローラ11は、数11のXを時間経過とともに順次演算し、位置制御データ(X)を作成する。この位置制御データ(X)は、サーボコントローラ12に供給され、これにより、数11のXに対応した励磁電流がソレノイド5に供給される。この場合、サーボコントローラ12は、ソレノイド5のフィードバック信号と位置制御データ(X)とを比較し、両者が一致するように励磁電流を制御するから、ソレノイド5のプランジャの突出量は、数11のXに対応したものとなる。したがって、鍵1は数11で示される放物線軌道に従って押下されていき、リファレンスポイント X_r においてリファレンス速度 V_r および加速度 a を有する運動を行う。この結果、記録時の打弦強度を忠実に再現

する打弦が行われる。しかも、人間の演奏状態に近い放物線軌道により、加速度をも再現するので、演奏の微妙なニュアンスも再現される。また、押鍵軌道と離鍵軌道が交差するときは、再生前処理部10が数22によって交差時刻 t_c を演算し、この時刻を境に、押鍵軌道と離鍵軌道を切り換えてモーションコントローラ11に供給する。これにより、鍵1は放物線の押鍵軌道の途中から直線の離鍵軌道に、あるいは直線の離鍵軌道の途中から放物線の押鍵軌道に切り換えられ、いわゆる、ハーフストロークの奏法に従った運動を行う。

【0110】ここで、図12に本実施例によって鍵駆動した場合の実験例を示す。この図12に示す3つのグラフは、いずれも横軸が時間であり、縦軸が鍵のレスト位置 X_0 からの移動量を示している。この図に示す(イ)は、演奏者が実際に演奏を行った場合の鍵の軌道を示しており、図に示す時刻 t_{10} 、 t_{11} の付近はハーフストローク奏法のために、鍵がレスト位置 X_0 に戻る前に次の押鍵動作に入っている。また、時刻 t_{12} 付近は、鍵は動いているが打弦は行われなかった部分である。また、図12(イ)に示す演奏から得られる打弦時刻、打弦速度、キーオン時刻、キーオン速度、離鍵時刻および離鍵速度を基にして、本実施例において再現した押鍵軌道および離鍵軌道が同図(ハ)に示す軌道である。また、同図(ロ)は、同図(イ)および(ハ)に示す2つの軌道を重ねたものであり、両者が良く一致していることが判る。例えば、ハーフストロークの部分については、再生軌道もハーフストローク軌道になっており、打弦が行われなかった時刻 t_{12} の部分については、再生軌道は鍵を押し切った状態が継続されている。

【0111】また、図13は、図12の一部を拡大したものであり、この図からも実際の演奏による鍵軌道と、実施例において再現した鍵軌道との一致性が確認される。なお、実際の演奏による鍵軌道に対して、実施例で再現した鍵軌道は揺らぎが少なく一見単純に見えるが、図12(イ)の軌道に含まれる揺らぎは、演奏者の意図とは違った部分(ミスによる部分)がほとんどである。したがって、実施例による軌道は、ミス部分を除外し、理想に近づいた軌道であると言える。

【0112】C. 第2実施例の変形

①ダンパー離弦タイミングの再現

鍵を押下していくと、まず、ダンパー6(図1参照)が弦から離れ、次いで、ハンマー2が打弦を行うことは周知の通りである。この場合、ダンパー6が弦から離れても、打弦が行われるまでは音は発生しないから、ダンパー6の離弦を正確に再現しても意味がないという考えもある。しかしながら、複数の鍵を同時あるいは近接したタイミングで押鍵する場合などには、他の弦の振動や共鳴の影響のために、打弦前であっても、ダンパーが離れていれば、その弦が振動することがある。したがって、演奏における他の弦の影響を忠実に再現するために、ダ

ンバーの離弦タイミングを正確に再現することも重要であると言える。そこで、以下のようにして、ダンパー離弦タイミングを推定する。まず、前述のように、キーオン時刻を t_{k_1} 、キーオン速度を V_k とし、キーオン位置（鍵1に設けられたシャッタ26がキーセンサ25内の下方のフォトセンサを遮光したときの鍵の位置）を X_{k_1} とする。次に、図14に示すように、ダンパーの離弦に対応する鍵の位置を X_1 とし、キーオン速度 V_k の傾きに沿って直線を引く。そして、この直線と位置 X_1 との交点の時刻をダンパー離弦時刻 t_1 と推定する。すなわち、ダンパー離弦時刻 t_1 は次式によって求められる。

【0113】

【数26】 $t_1 = t_{k_1} - (X_{k_1} - X_1) / V_k$

以上のようにして、ダンパー離弦時刻 t_1 が求められれば、前述の各実施例と同様にしてリファレンス速度 V_r およびリファレンス時刻 t_r を求め、これらを満たす放物線軌道を求める。この放物線軌道を図15に示す。この軌道は、リファレンス時刻 t_r とリファレンスポイント X_r を原点にとれば、次のように表される。

【0114】

【数27】 $X = ((X_1 - V_r \cdot (t_1 - t_r)) / (t_1 - t_r)^2) \cdot t^2 + V_r \cdot t$

したがって、数27に示される軌道にしたがって、ソレノイドの励磁電流を制御すれば、ダンパー離弦時刻を正確に再現することができる。なお、励磁電流の制御は、前述した各実施例と同様に行えばよい。

【0115】②加速度、初速度の固定

第2実施例においては、押鍵時の加速度 a と初速度 V_0 の双方を演算によって求めるようにしたが、いずれか一方を固定値にして、計算を簡略化してもよい。この場合の固定値は、実験等を繰り返すことによって、統計的に適切な値を決めることが望ましい。例えば、加速度を a を固定する場合には、 $a = 2.5 \text{ m/s}^2$ に設定すると、比較的良好な結果が得られることが判った。ここで、 $a = 2.5 \text{ m/s}^2$ にした場合の実験結果を図16に示す。

【0116】この図に示す(イ)は、演奏者が実際に演奏を行った場合の鍵の軌道を示しており、(ハ)は同図(イ)に示す演奏から得られる打弦時刻、打弦速度を基にするとともに、加速度を固定して再現した押鍵軌道を示す。すなわち、前述した数11に示す軌道における加速度 a の値を固定し、数12に示す加速度算出を省略して押鍵軌道を求めたものである。なお、離鍵軌道は、直線軌道であるため、前述の実施例と同様である。また、同図(ロ)は、同図(イ)および(ハ)に示す2つの軌道を重ねたものであり、両者が良く一致していることが判る。例えば、ハーフストロークの部分についても、良好な一致性が認められる。なお、図17(イ)、

(ロ)、(ハ)は、図16(イ)、(ロ)、(ハ)の軌

道を一部拡大したものである。以上のように加速度を固定すると、演算が省略できるため、処理速度が早くなる。

【0117】次に、初速度 V_0 を固定した場合について説明する。この場合には、数21の演算を省略し、予め設定した初速度 V_0 を用いて数11に示す軌道を演算すればよい。ここで、初速度 V_0 を 0.1 m/s にした場合の実験結果を図18に示す。この図に示す(イ)は、演奏者が実際に演奏を行った場合の鍵の軌道を示しており、(ハ)は同図(イ)に示す演奏から得られる打弦時刻、打弦速度を基にするとともに、初速度 V_0 を固定して再現した押鍵軌道を示す。すなわち、前述した数21の初速度演算を省略して初速度 V_0 を固定し、この条件の基で数11に示す押鍵軌道を求めたものである。なお、離鍵軌道は、直線軌道であるため、前述の実施例と同様である。そして、同図(ロ)は、同図(イ)および(ハ)に示す2つの軌道を重ねたものであり、両者が良く一致していることが判る。なお、図19(イ)、(ロ)、(ハ)は、図18(イ)、(ロ)、(ハ)の軌道を一部拡大したものである。

【0118】III. 第3実施例

A. 実施例の制御原理

①リアルタイムの再生

上記第1、第2実施例においては、まず演奏記録部30において各種鍵イベントおよびハンマーイベントを記録しておき、演奏記録が終了した後に記録後処理部31による正規化処理、あるいはダミーイベントの追加等が行われていた。しかし、演奏データをリアルタイムに出力しなければならない場合もある。例えば、演奏者による自動ピアノの生演奏を行いながら演奏データを検出し、これによって別室の他の自動ピアノを駆動することも考えられる。このような場合、データ送信側自動ピアノにおける演奏記録部30と記録後処理部31、および受信側自動ピアノにおける再生前処理部10の処理をリアルタイムに実行する必要がある。しかし、これを実現するためには、以下のような点について留意する必要がある。

【0119】②データ遅延の必要性

上記第1、第2実施例においては、押鍵速度や打弦速度等に基づいてリファレンス軌道を生成し、これに基づいて各鍵を駆動していたため、少なくとも押鍵速度または打弦速度のうち一方が得られなければリファレンス軌道を生成できず、鍵を駆動することができない。本実施例においてもリファレンス軌道を求める原理は第1、第2実施例と共通するから、受信側自動ピアノの演奏タイミングは送信側自動ピアノのものと比較して若干遅延せざるを得ない。

【0120】すなわち、送信側自動ピアノにおいて鍵1が押鍵されると、やがて鍵1はセンサ位置に達し、押鍵時刻と押鍵速度が検出される。その後、ハンマー2がセ

ンサSE1、SE2を通過すると、打弦時刻と打弦速度が検出される。これらのデータに基づいて押鍵時のリファレンス軌道が生成された後、初めて受信側自動ピアノの鍵の駆動を開始することができるのである。また、ダミーイベントが有る場合にはイベントの内容を修正する処理を行うので、各押鍵鍵に対応して行われるリファレンス計算に要する時間もダミーイベントの有無によって各押鍵鍵で異なる。そこで、本実施例においては、各イベント毎にタイムコード（時刻情報）を付与し、各イベント間の相対的な時間差を保持しつつ演奏データを遅延させる。このときの遅延時間は情報の最大処理時間、すなわち押鍵開始時刻から打弦時刻までの最大時間とリファレンス軌道の計算時間と記録後処理部31の処理時間との合計値である。以下、前二者の内容について各々について説明しておく。

【0121】(i)押鍵開始時刻から打弦時刻までの最大時間

ここで、押鍵開始時刻から打弦時刻までの最大時間とは、ステップSP106（図6参照）で説明した値“B／押鍵速度”よりも若干大きい値である。すなわち、ステップSP106で比較対象にされている「時間差」とは押鍵時刻から打弦時刻までの時間であって、押鍵開始時刻から打弦時刻までの時間は、さらに長くなるからである。この最大値を決定するためには、押鍵速度の最小値を予め決定する必要がある。すなわち、押鍵速度の最小値は発音されない程度の速度にしておき、実際に検出された押鍵速度が最小値よりもさらに小さい場合には、押鍵速度を強制的に該最小値に設定するとよい。

(ii)リファレンス軌道の最大計算時間

リファレンス軌道の計算は、複数の鍵について同時に行われる場合もあるため、一つの押鍵イベントに要する処理の数倍の時間を想定しておくといよい。

【0122】③イベントデータの並び替えの必要性

自動演奏を再生する場合には、タイムコードの小さい順に各イベントを再生する必要がある。このことは、一般的な自動ピアノでは、検出された順序で各イベントを記録しておき、記録した順序で再生することにより、自然に解決される。

【0123】しかし、本実施例のようにリファレンス軌道を生成する場合には、検出時のイベント順序と比較して、再生時のイベント順序は異なる。このことを図26（イ）～（ハ）を参照し説明する。同図（イ）において、時刻 t_1 にキーコードKC1の押鍵イベントが検出されており、その後、時刻 t_2 にキーコードKC2の押鍵イベントが検出されている。これらキーコードKC1、2のリファレンス軌道は同図（ロ）に示すようになるが、キーコードKC2の押鍵開始時刻 t_{21} はキーコードKC1の押鍵開始時刻 t_{12} よりも早くなっている。従って、リファレンス軌道を押鍵開始等のイベントによって表現するならば、キーコードKC1、2のイベ

ントの順序が逆転することになる。

【0124】従って、処理を行う順番で各イベントを配置するために、イベントデータの順序を並び替える必要が生じる。ここで、各イベントデータの内容としては、キーコード、押鍵開始時刻（タイムコード）、押鍵初速度、加速度等の内容を含める必要があり、メモリ上でこれらのデータ自体を入れ替えるのは多大な処理時間を要する。そこで、本実施例にあっては、メモリ上のイベントデータを構造体によって表現し、前後のイベントに係る構造体のアドレスを指定するポインタ変数を、この構造体のメンバとして含ませることにより、イベントの並び替えを高速に行うこととしている。

【0125】その詳細を図24（イ）～（ハ）を参照し説明する。同図（イ）においてST₁～ST_nは構造体であり、 x_1 ～ x_n はメモリ上における各構造体の先頭アドレスである。構造体ST₁～ST_nは、各々一つのイベントに係るデータ（キーコード、押鍵開始時刻等）と、ポインタ*PRIOR、*NEXTをメンバとして含んでいる。ここで、構造体ST₁～ST_nの順序で各イベントを実行する場合には、各構造体のポインタ*NEXTには、次に実行すべきイベントに係る構造体の先頭アドレスが格納される。すなわち、構造体ST₁のポインタ*NEXTには構造体ST₂の先頭アドレスであるアドレス x_2 が格納され、同様に構造体ST₂のポインタ*NEXTにはアドレス x_3 が格納されている。また、最後に実行されるイベントに係る構造体ST_nのポインタ*NEXTには、最初に実行されるイベントに係る構造体ST₁の先頭アドレス x_1 が格納されている。

【0126】また、各構造体のポインタ*PRIORには、先に実行すべきイベントに係る構造体の先頭アドレスが格納されている。但し、最初に実行すべきイベントに係る構造体ST₁のポインタ*PRIORには、最後に実行されるイベントに係る構造体ST_nの先頭アドレス x_n が格納されている。かかるデータ構造においては、各構造体はポインタ*PRIOR、*NEXTによってリンクされており、各構造体のポインタ*PRIOR、*NEXTを順次追跡していくことにより、昇順または逆順に各イベントに係る構造体を高速にアクセスすることができる。

【0127】ここで、何れかのイベントを削除する場合は、そのイベントに係る構造体をリンクから外してしまえば良い。同図（ロ）に示す例にあっては、構造体ST₂のポインタ*NEXTがアドレス x_3 に変更され、構造体ST₂のポインタ*PRIORがアドレス x_1 に変更されており、これによって構造体ST₂がリンクから外されている。すなわち、ここに言う「削除」とは、上述のように何れかの構造体をリンクから外すことであり、削除された構造体の内容自体は、実際に消去されない限りメモリ内に残存することになる。

【0128】また、任意の箇所になんらかのイベントを追加する場合は、挿入箇所におけるポインタ*PRIOR、*

NEXT を変更すればよい。同図(ハ)は、先頭アドレス x_{n+1} に記憶されている構造体 ST_{n+1} をリンクに組込む例を示している。図において構造体 ST_1 のポインタ *NEXT および構造体 ST_2 のポインタ *PRIOR は、各々アドレス x_{n+1} に変更されており、構造体 ST_{n+1} のポインタ *PRIOR はアドレス x_1 に、ポインタ *NEXT はアドレス x_2 に各々設定されている。これにより、構造体 ST_{n+1} は、上記リンクにおいて、構造体 ST_1 と構造体 ST_2 の間に追加される事になる。イベントの順序を変更する場合は、移動したいイベントに係る構造体を削除し、削除したイベントを所望の箇所に追加すればよい。

【0129】B. 実施例の構成

次に、本実施例の構成を図27を参照し説明する。図において100、200は自動ピアノであり、図1のものと同様に構成されている。300はリアルタイム処理装置であり、CPU、RAM、ROM等によって構成されている。リアルタイム処理装置300は、後述する処理プログラムに基づいて、図1における演奏記録部30、記録後処理部31および再生前処理部10と同等の処理をリアルタイムに実行するものである。すなわち、リアルタイム処理装置300は、自動ピアノ100内の各キーコードに対応するセンサSE1、SE2、25に接続され自動ピアノ100における各種イベントを検出するとともに、これらのイベントに基づいてリファレンス軌道を生成し、自動ピアノ200内のモーションコントローラ11を介して自動ピアノ200を駆動する。

【0130】C. 実施例の動作

C-1. リアルタイム処理装置300の動作

次に、本実施例の具体的な動作を説明するが、最初にリアルタイム処理装置300の動作について詳述しておく。まず、リアルタイム処理装置300の電源を投入すると、図25に示す処理プログラムが起動される。図において処理が開始されると、ステップSP1において所定の初期設定が行われる。このステップSP1においては、所定の時刻レジスタが「0」に設定される。ところで、リアルタイム処理装置300においては「1msec」毎にタイマ割込が発生し、図28に示す割込処理ルーチンが起動され、上記時刻レジスタが「1」だけインクリメントされる。従って、この時刻レジスタは、ミリ秒単位でリアルタイム処理装置300内の絶対時刻を表示することになる。

【0131】次に、図25において処理がステップSP2に進むと、時間監視キーがタイムアップしたか否かが判定される。ここで、時間監視キーとは、押鍵イベントが検出される毎にメモリ内に設けられ、該押鍵イベントに対するハンマーイベントが発生すべき最も遅い時刻を記憶しておく変数である。この時点では時間監視キーは設定されていないから「NO」と判定され、処理がステップSP4に進む。ステップSP4においては、自動ピ

アノ100から新たなイベントデータを受信したか否かが判定される。新たなデータを受信していなければ「NO」と判定され処理がステップSP12に進む。

【0132】ステップSP12においては、現在時刻が出力時刻(詳細は後述する)を超過したか否かが判定される。この時点では出力時刻が全く設定されていないため「NO」と判定され、処理がステップSP2に戻る。以後、なんらかのイベントデータが入力されるまで、ステップSP2、4、12から成るループが繰り返されることになる。

【0133】次に、実際にイベントデータが入力された場合の処理を図26を参照して説明する。同図(イ)においては、時刻 t_1 にキーコードKC1の押鍵イベントが発生し、リアルタイム処理装置300に受信される。このイベントが検出された後、処理がステップSP4に進むと、ここで「YES」と判定され処理がステップSP5に進む。ステップSP5においては、受信したイベントに係る構造体がメモリ内に生成される。すなわち、この構造体は、キーコード、押鍵時刻(この時点における時刻レジスタの内容)、押鍵速度、およびポインタ *PRIOR、*NEXT等をメンバとするものであり、ポインタ *PRIOR、*NEXT にはその構造体自体の先頭アドレスが格納される。

【0134】なお、この押鍵イベントはリアルタイム処理装置300が動作した後最初に受信されたイベントであるため、既存のイベントとリンクされることは無いが、以後受信されるイベントは、構造体としてメモリに記憶される際に、順次既存の受信イベントとリンクされる。このリンクを以後「記録フレーム」という。また、ステップSP5においては、この押鍵イベントに対応する時間監視キーがメモリ内に設けられる。その値は、時刻 t_1 と、「押鍵開始時刻から打弦時刻までの最大時間」との合計値である。これにより、以後ステップSP2が実行される毎に時間監視キーが参照され、これがタイムアップすると、ステップSP3を介して、ダミーイベントが記録フレームに追加されることになる。

【0135】次に処理がステップSP6に進むと、完成したイベントが記録フレーム内に生じたか否かが判定される。ここで「完成したイベント」とは、リファレンス軌道を生成するに足るイベントである。上記時刻 t_1 における押鍵イベントは、後のハンマーイベントと組み合わせることによって初めてリファレンス軌道を生成できるものであるから、この時点では完成していない。従って、ここでは「NO」と判定され、処理がステップSP12に進む。以後、ステップSP2、4、12から成るループが繰り返されることになる。

【0136】次に、図26の時刻 t_2 において、自動ピアノ100でキーコードKC1のハンマーイベントが発生すると、次にステップSP4が実行された際に「YES」と判定され、処理がステップSP5に進む。ステッ

ブSP5においては、ハンマーイベントの内容が構造体としてメモリ内に記憶され、この構造体は記録フレームにリンクされる。これにより、キーコードKC1については押鍵イベントとハンマーイベントが揃ったため、リファレンス軌道を求めることが可能になった。すなわち、両者は「完成したイベント」になった。

【0137】従って、次に処理がステップSP6に進むと「YES」と判定され、処理がステップSP7に進む。ステップSP7においては、完成したイベント内にダミーイベントを含むかが判定される。上記例においてはダミーイベントは含まれないため「NO」と判定され、処理がステップSP10に進む。ステップSP10においては、押鍵開始時刻、交差点あるいは加速度等についてリファレンス計算が行われ、その結果が構造体としてメモリ内に格納される。以後、リファレンス計算の結果は、記録フレームと同様に、構造体としてメモリ内に記憶されるとともに押鍵開始時刻の早い順にリンクされる。このようにリファレンス計算の結果（構造体）をリンクしたものを、以後「リファレンスフレーム」という。

【0138】ところで、時刻 t_{i-1} 、 t_i におけるイベントに基づいて押鍵開始時刻を演算すると、例えば同図（ハ）に示す時刻 t_{i-1} が得られる。この時刻 t_{i-1} は、演算の完了した時点では既に過去の時刻であるから、時刻 t_{i-1} を押鍵開始時刻とするようなリファレンス軌道を生成することは不可能である。また、検出された各イベントについてリファレンス計算することにより、検出時の各イベント間の時間間隔とリファレンス計算後の各イベント間の時間間隔が異なることがあり、最たる場合には、前述したように、後に検出されたイベントを先に検出されたイベントよりも早く再生する必要がある。従って、キーコードKC1の押鍵開始時刻は、時刻 t_{i-1} から所定の遅延時間 t_d だけ遅らせた時刻 t_i に設定される。すなわち、実際のリファレンスフレームにおいては、同図（ニ）に従って押鍵開始時刻等が設定される。

【0139】次に、処理がステップSP12に進むと、リファレンスフレーム内の先頭にある構造体について、出力時刻が超過したか否かが判定される。この時点では未だ時刻 t_i に達していないため、「NO」と判定され処理がステップSP2に戻る。以下同様に、自動ピアノ100において新たなイベントが生ずる毎に、その内容が記録フレームに追加され、「完成したイベント」が生ずると、その内容に基づいて押鍵開始時刻、交差点あるいは加速度等についてリファレンス計算が行われ、その結果がリファレンスフレームに追加される。

【0140】次に、時刻が t_i に達した後に処理がステップSP12に進むと、ここで「YES」と判定され、処理がステップSP13に進む。ステップSP13においては、出力時刻の超過したリファレンスフレーム

内の構造体の内容に基づいて、リファレンス軌道が生成され、その内容が自動ピアノ200のモーションコントローラ11に供給される。リファレンス軌道がモーションコントローラ11に供給されると、記録フレームおよびリファレンスフレーム内の対応するイベントデータは用済になるため、各々のフレームから削除される。削除されたイベントデータに係るメモリ領域は解放され、あらたなデータを格納することが可能になる。

【0141】ところで、上述のようにステップSP2において「時間監視キーがタイムアップした」と判定されるとダミーイベントが追加されるのであるが、この場合の処理について詳述しておく。ダミーイベントが追加された場合は、基本的には、ダミーイベントが追加されなかった場合と同様の処理が行われる。すなわち、ダミーイベントに基づいてリファレンス計算が行われる。しかし、ステップSP3でダミーイベントが追加された直後のステップSP4において、本来の（ダミーではない）イベントが受信されると、イベントが重複するおそれがある。このようなケースは稀ではあるが、本実施例においてはかかる場合も不具合が生じないように対応している。

【0142】まず、ダミーイベントを含む「完成したイベント」が生じた場合、ステップSP7において「YES」と判定され、処理がステップSP8に進む。ステップSP8では当該ダミーイベントを採用するか否かについて最終的な判断が行われる。すなわち、上述したように本来のイベントを受信した場合には、その受信したイベントを用いることが好適であるから、かかる場合には処理をステップSP9に進め、ダミーイベントを記録フレームから削除することとしたものである。

【0143】C-2. 演奏再生の全体動作

次に、演奏再生時における実施例の全体動作を説明する。まず、演奏者が自動ピアノ100を演奏すると、この演奏による鍵イベントおよびハンマーイベントが自動ピアノ100内のセンサSE1、SE2、25によって検出され、押鍵速度、あるいは離鍵速度とともにリアルタイム処理装置300に供給される。リアルタイム処理装置300においては、受信した各イベントの内容が記録フレームに追加され、イベントの完成とともにリファレンス計算が行われると、その計算結果がリファレンスフレームに追加される。

【0144】そして、リファレンスフレーム内の各イベントについて出力時間が超過すると、リファレンス軌道が演算され、その結果が自動ピアノ200のモーションコントローラ11に供給される。まず、押鍵軌道データがモーションコントローラ11に供給されると、この押鍵軌道データは位置制御データ（X）に変換される。すなわち、数3に示される軌道データがモーションコントローラ11に供給され、モーションコントローラ11は、数3のXを時間経過とともに順次演算し、位置制御

データ(X)を作成する。

【0145】この位置制御データ(X)は、サーボコントローラ12に供給され、これにより、数3のXに対応した励磁電流がソレノイド5に供給される。この場合、サーボコントローラ12は、ソレノイド5のフィードバック信号と位置制御データ(X)とを比較し、両者が一致するように励磁電流を制御するから、ソレノイド5のプランジャの突出量は、数3のXに対応したものとなる。したがって、鍵1は数3で示される直線軌道に従って押下されていき、リファレンスポイントX_rにおいてリファレンス速度V_rを有する運動を行う。これにより、自動ピアノ100における打弦強度は、自動ピアノ200においてリアルタイムに、かつ忠実に再現される。

【0146】次に、リアルタイム処理装置300は、リファレンスフレーム内の離鍵終了時刻データ、離鍵速度データに応じて、離鍵軌道を作成し、自動ピアノ200のモーションコントローラ11に供給する。モーションコントローラ11は、上述の場合と同様にして、数6のX_Nを時間の経過に応じて順次演算し、位置制御データ(X)を作成し、サーボコントローラ12に供給する。この結果、サーボコントローラ12は数6に対応した励磁電流制御を行い、鍵1は数6で示される直線軌道に従ってレスト位置X₀に戻っていく。これにより、自動ピアノ200における鍵1の運動は、離鍵リファレンスポイントX_{rN}において離鍵リファレンス速度V_{rN}を有するものとなり、自動ピアノ100の離鍵状態が忠実に再現される。

【0147】以上のように、本実施例によれば、イベントに時刻情報を含めるとともにリファレンス軌道のイベントを所定時間遅延させるようにしたので、後に検出されたイベントを先に検出されたイベントよりも早く再生する必要がある場合など、検出時の各イベント間の時間間隔とリファレンス計算後の各イベント間の時間間隔が異なっても、リファレンス計算後の各イベント間の時間間隔に応じた再生を行うことができる。また、本実施例によれば、イベントに時間情報を含めるとともにリファレンス軌道のイベントを所定時間遅延させるようにしたので、ダミーイベントの有無によってイベント毎に処理時間が異なっても、リファレンス計算後のイベント間の時間間隔は保持される。

【0148】なお、イベントを所定時間遅延させてイベント毎の処理時間を吸収する手法は、上述したダミーイベントの有無によるリファレンス計算以外にも適用可能であって、要するに、イベント毎に処理時間が異なるものであれば適用可能である。例えば、リファレンス計算において、押鍵イベントについては加速度の計算を行い、離鍵イベントについては加速度計算を行わない場合には、両者における計算時間は当然異なる筈である。また、リファレンス計算に直接関係の無い用途としては、

例えばキーコードに応じて打弦速度を正規化するような処理(低音域と高音域の打弦速度は少し大きくし中音域の打弦速度は変更せずに出力すること)が挙げられる。

【0149】IV. 変形例

この発明においては、上述した各実施例に対し、以下に述べる種々の変形が可能である。

A. サーボ制御の変形など

上述した各実施例においては、モーションコントローラ11とサーボコントローラ12によって位置サーボ制御を行っていたが、これに代えて、第1実施例では速度を指示する速度サーボ制御を行っても良い。すなわち、モーションコントローラ11が押鍵(あるいは離鍵)開示時刻において初速度を指示し、サーボコントローラ12は鍵1が与えられた速度を保つようにサーボ制御を行うようにしてもよい。また、第2実施例においては、加速度制御を行うようにしてもよい。すなわち、モーションコントローラ11が押鍵開始時刻において初速度を指定し、以後は時間の経過とともに速度の変化分を順次指示する。そして、サーボコントローラ12は、速度変化分を累算して、現時点の速度を算出し、鍵1の速度が累算速度に一致するようにサーボ制御を行う。

【0150】B. 推定した加速度情報の電子楽器への応用

第2実施例において推定した加速度aは、電子ピアノの押鍵制御に限らず、電子楽器等の楽音制御に用いることができる。すなわち、加速度の推定は、押鍵速度情報と打弦速度情報がそれぞれ1以上あれば可能であるから、これらの情報を基に加速度を推定し、それを楽音制御情報として用いることができる。例えば、楽音波形を複数記憶し、これらの波形を音の強さや、高さに応じて選択して読み出す電子楽器があるが、この電子楽器に、さらに加速度情報に応じた波形選択を行わせると、発生楽音の表情がいっそう豊かになるという利点が得られる。

【0151】C. リファレンスポイントX_rにおける運動属性の変形

上述した各実施例においては、リファレンスポイントX_rにおける鍵の速度や加速度を再現するようにしたが、鍵の運動を一意的に再現できるのであれば、その他の運動属性を再現するようにしてもよい。例えば、速度、加速度、力のいずれか、あるいは、これらの組み合わせを用いても良い。

【0152】D. 離鍵時の加速度軌道

上述した各実施例においては、離鍵時の軌道は全て直線軌道であったが、これを加速度軌道としてもよい。この場合には、離鍵速度を2カ所以上で測定するようにセンサを追加し、得られた速度の変化から加速度を推定するように構成すればよい。例えば、キーセンサ25内のフォトセンサの数を増やし、演奏記録部30がこれらのセンサの出力信号から2カ所以上の離鍵速度を検出するようにすればよい。さらに、押鍵時と離鍵時の軌道を直線

とするか放物線とするかは、任意に組み合わせることが可能である。

【0153】E. 開平演算態様

第2実施例における初速算出演算(数20)および押鍵開始時刻算出演算(数18、19)は、開平演算が入るが、ソフトウェア上の演算速度が問題になる場合は、例えば、2分探索法を用いて、整数演算をしてもよい。

【0154】F. 弱音の再生

実際の押鍵操作においては、強打よりも弱打において放物線軌道をとる傾向が強く、直線軌道での再現では再現性が悪化してくる。特に、最弱打(pppp)に近い押鍵を直線軌道で再現しようとする、ピアノのアクション部の構造などの理由から、ハンマーへの力伝達が行われず不安定な打弦となることがある。そこで、第1実施例においては、弱音だけを検出して放物線軌道で再現するようにし、その他の音は直線軌道で再現するように構成してもよい。すなわち、再生前処理部30において、打弦速度データを所定のしきい値で振り分け、弱音と判定された音に対しては、予め設定した加速度aを用いて放物線軌道を算出し、これによって鍵を駆動する。

【0155】G. 打弦時刻、打弦速度の代用

例えば、特開平1-239594号では、押鍵速度から打弦速度を推定して記録するようにしているが、このように、打弦速度に代わるデータが記録されている場合は、前述した各実施例において、それを用いることも可能である。したがって、MIDI信号中のキーオン速度(キーベロシティ)のように、打弦速度に対応するデータが記録されている場合においては、これを用いて軌道演算を行うことができる。打弦時刻についても、これに代わるデータがあれば、代用することができる。要は、発音時刻や発音強度に関連したデータであれば、用いることができる。

【0156】H. 記録フレーム、リファレンスフレームの代用

第3実施例では構造体内のポイントをを用いて各構造体をリンクし記録フレーム、およびリファレンスフレームを構成したが、各構造体をリンクする手段は構造体内のポイントに限られない。例えば、磁気ディスク記録に用いられているFAT(ファイル・アロケーション・テーブル)と同様な技術を用いることもできる。すなわち、メモリ内の所定領域において、タイムコード順にアドレス情報を順次配置しておき、このアドレス情報を参照することによって各種データをアクセスしてもよい。

【0157】I. ダミーイベント追加条件の変更

図6においては、ステップSP106、110を省略してもよい。すなわち、ステップSP105または109において「YES」と判定されると直ちにステップSP107または111を実行するようにしてもよい。

【0158】J. 第1実施例においては、演奏データを外部に出力する(記録媒体に記録する)前に抜けイベン

トの修復を行っていたが、要はリファレンス計算を行う前に抜けイベントの修復を行えばよいのであって、センサ出力を何ら修正することなく記録媒体に記録し、これを読み出して自動演奏をする際に抜けイベントの修復を行うようにしてもよい。また、同様に、第3実施例において、自動ピアノ100あるいは自動ピアノ200にリアルタイム処理装置300を内蔵して自動ピアノ100あるいは自動ピアノ200のどちらか一方でイベントの修復処理とリファレンス計算を行うようにしてもよいし、自動ピアノ100でイベントの修復処理のみを行い、自動ピアノ200でリファレンス計算のみを行うようにしてもよい。

【0159】K. ダミーイベントの内容を修正する際の態様は、上述した態様に限らず、どのような態様であってもよい。例えば、上述した第1実施例においては、離鍵/押鍵イベントが抜けた場合に、図21の直線L₁に示すように、ダミーの離鍵速度は最大の離鍵速度に設定されるようにしたが、図21の二点鎖線L₂に示すように、ダミーの押鍵イベントの押鍵時刻に向けて離鍵させたときの速度にしてもよい。

【0160】L. 第3実施例においては、入力されるイベントの時間間隔に応じてダミーイベントを追加するようにしたが、これに限らず、第1実施例と同様に、押鍵イベント、ハンマーイベント、離鍵イベントの入力される順序に応じて離鍵/押鍵イベント抜け、ハンマーイベント抜け等を検出してダミーイベントを追加するようにしてもよい。

【0161】M. 上記各実施例において開示したイベント修復技術は、リファレンス計算にのみ適用されるものではなく、他の種々の用途に用いて良いことは言うまでも無い。例えば、特開平1-239594号公報に開示されたもののように、イベントデータを用いて各種の処理を行う装置に対して、かかるイベント修復技術は適用可能である。

【0162】N. 第1実施例においては、ステップSP107、111において既存のイベントと同一の発生時刻を有する各種のダミーイベントが演奏データに追加され、ステップSP108、112においてこれらダミーイベントの発生時刻が修正されている。しかし、ステップSP107、111において、ダミーイベントを追加する際に、その発生時刻等を最初から適切な値に設定し、ステップSP107、111を省略してもよい。一方、第1実施例のようにダミーイベントの発生時刻を既存のイベントと同一時刻に設定しておく、そのこと自体によって当該イベントは例外的なイベントであることを表現でき、ダミーイベントと通常のイベントとを区別するための特別のコードを設ける必要がなくなる。

【0163】

【発明の効果】以上説明したように、請求項7に係る自動ピアノによれば、取得された限りの情報に基づいて正

確な鍵の動作再生を行うことができる。また、請求項1、4、5に係る演奏データ加工装置によれば、欠落したイベントが存在すると判定されると、演奏データに新たなイベントデータを適宜追加することができる。また、請求項2に係る演奏データ加工装置によれば、イベントデータ加工手段と遅延手段における処理時間を所定時間に保持することができるから、演奏データに対して、処理時間の異なる各種の処理を施した場合においても、各イベントデータ間の相対的な時間差を維持することが可能になる。さらに、請求項3、6に係る演奏データ加工装置によれば、各イベントデータに係るイベントの10 実行タイミングに基づいてイベントデータの遅延時間が設定されるから、各イベントデータの実行タイミングに応じて、各イベントデータを遅延手段から出力させることが可能になる。

【図面の簡単な説明】

【図1】 この発明の第1実施例の構成を示すブロック図である。

【図2】 リファレンスポイントを9.5mmに設定したときの鍵速度と打弦速度の関係を示す図である。

【図3】 リファレンス時間差 T_r と打弦速度との関係を示す図である。

【図4】 図3を縮尺2倍にした図である。

【図5】 図3を縮尺4倍にした図である。

【図6】 記録後処理部31のフローチャートである。

【図7】 鍵の押鍵軌道(直線軌道)を示す図である。

【図8】 数6で示される軌道を示す図である。

【図9】 押鍵の後に発生した離鍵の軌道が交差する場合を示す図である。

【図10】 第2実施例で用いる放物線軌道を示す図である。

【図11】 第2実施例における押鍵開始時刻の算出を説明するための図である。

【図12】 第2実施例において、鍵駆動した場合の実験例を示す図である。

*【図13】 図12の一部を拡大した図である。

【図14】 ダンパーの離弦時刻推定方法を説明するための図である。

【図15】 ダンパー離弦時刻を再現した場合の放物線軌道を示す図である。

【図16】 第2実施例において加速度を固定した場合の実験結果を示す図である。

【図17】 図16の一部を拡大した図である。

【図18】 第2実施例において初速度を固定した場合の実験結果を示す図である。

【図19】 図18の一部を拡大した図である。

【図20】 一般的な自動ピアノの構成を示す概略図である。

【図21】 第3実施例の動作説明図である。

【図22】 第3実施例の動作説明図である。

【図23】 第3実施例の動作説明図である。

【図24】 第3実施例のデータ構造の説明図である。

【図25】 リアルタイム処理装置300のメインルーチンのフローチャートである。

【図26】 リアルタイム処理装置300のタイムチャートである。

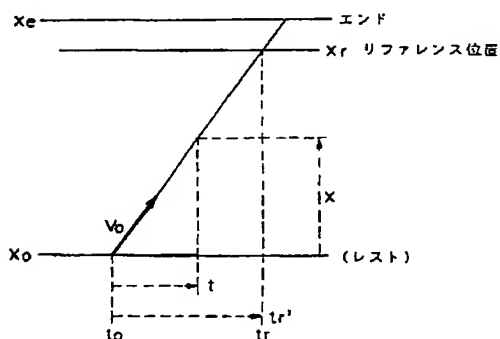
【図27】 この発明の第3実施例の構成を示すブロック図である。

【図28】 リアルタイム処理装置300のタイム割込処理ルーチンのフローチャートである。

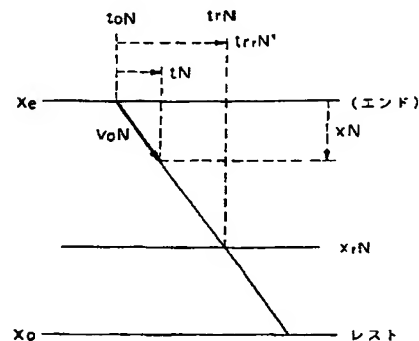
【符号の説明】

1……鍵、30……演奏記録部(記憶手段)、31……記録後処理部(判定手段、イベント修復手段)、10……再生前処理部(鍵駆動手段)、11……モーションコントローラ(鍵駆動手段)、12……サーボコントローラ(鍵駆動手段)、25……キーセンサ(イベント検出手段)、SE1、SE2……センサ(イベント検出手段)、300……リアルタイム処理装置(イベントデータ加工手段、演奏データ加工装置、遅延手段、遅延時間設定手段)。

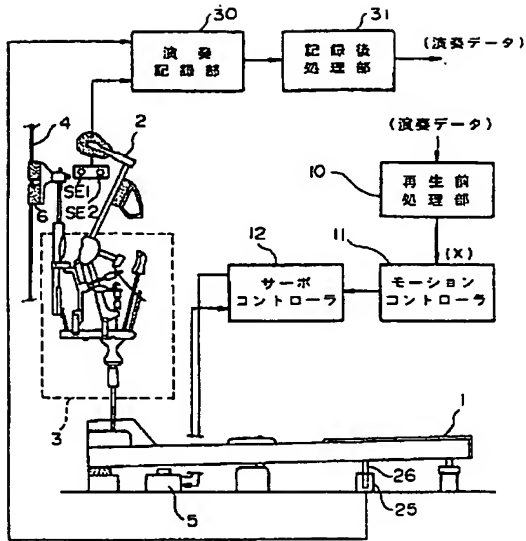
【図7】



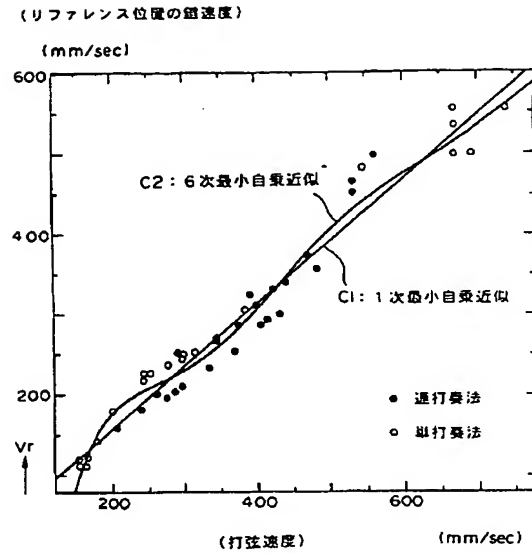
【図8】



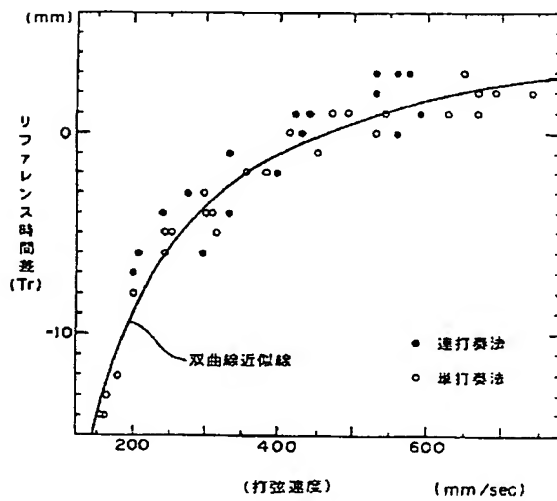
【図1】



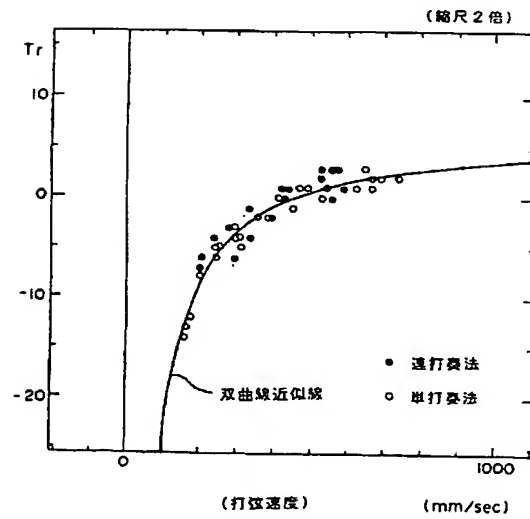
【図2】



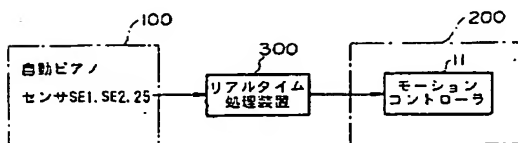
【図3】



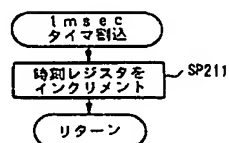
【図4】



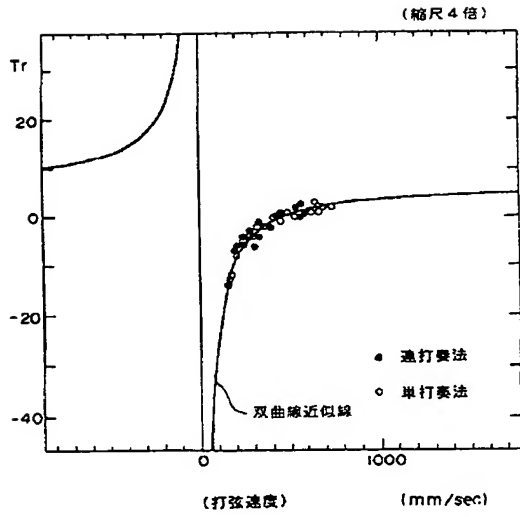
【図27】



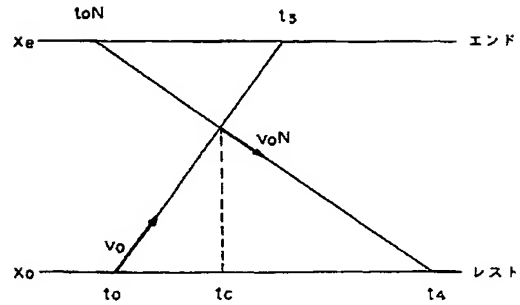
【図28】



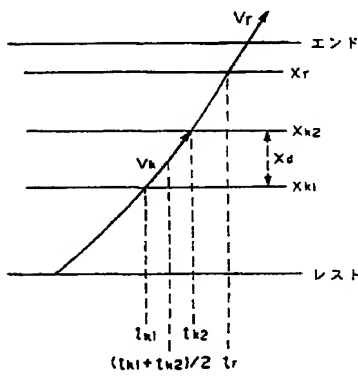
【図5】



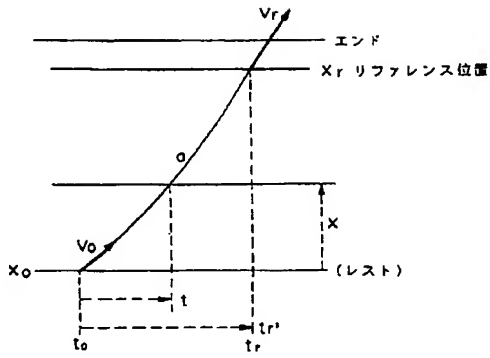
【図9】



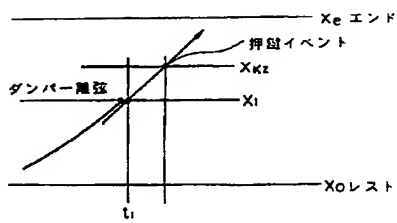
【図10】



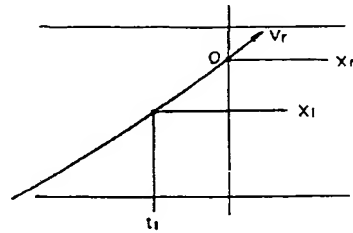
【図11】



【図14】

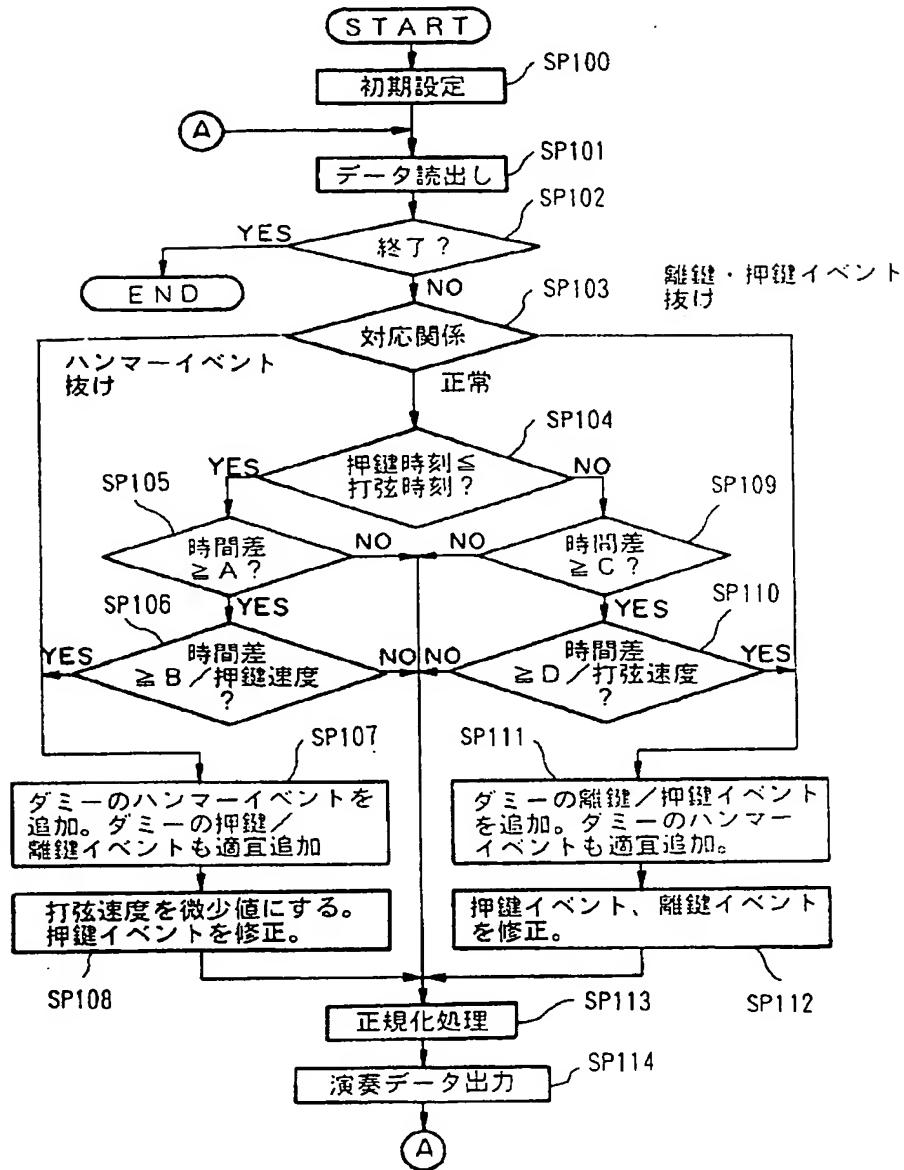


【図15】

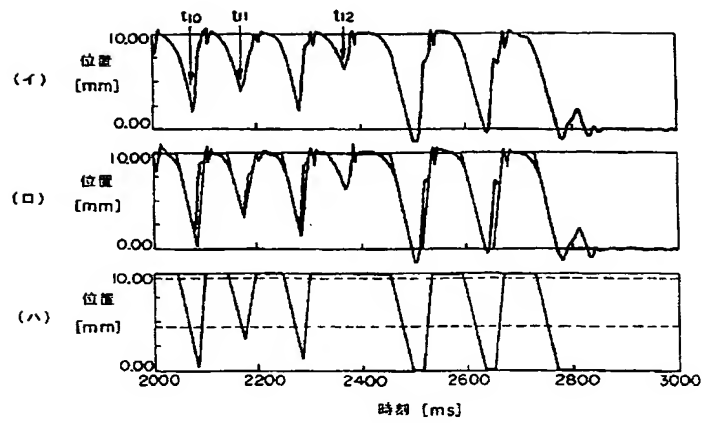


〔図6〕

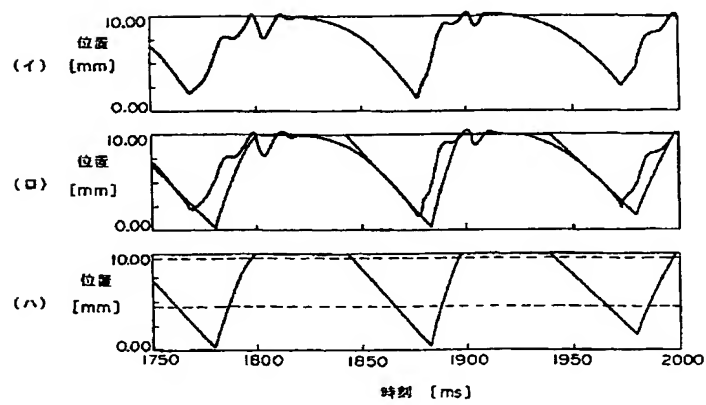
記録後処理部31の動作フローチャート



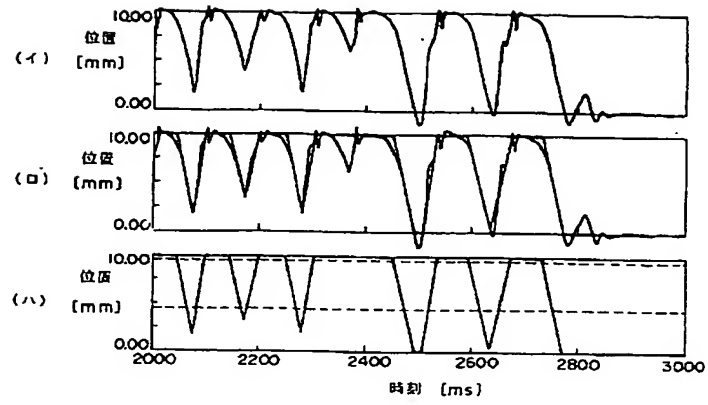
【図12】



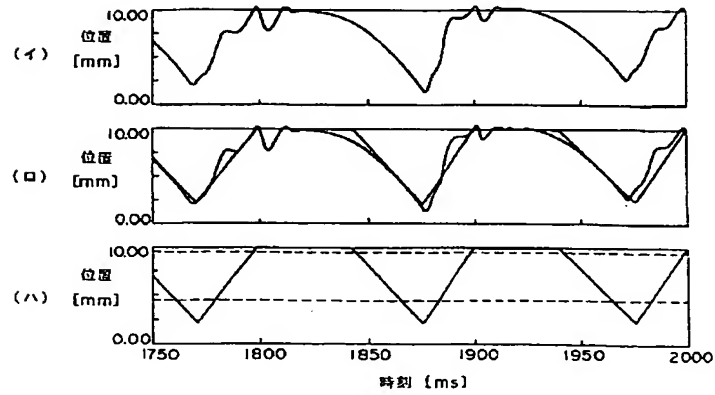
【図13】



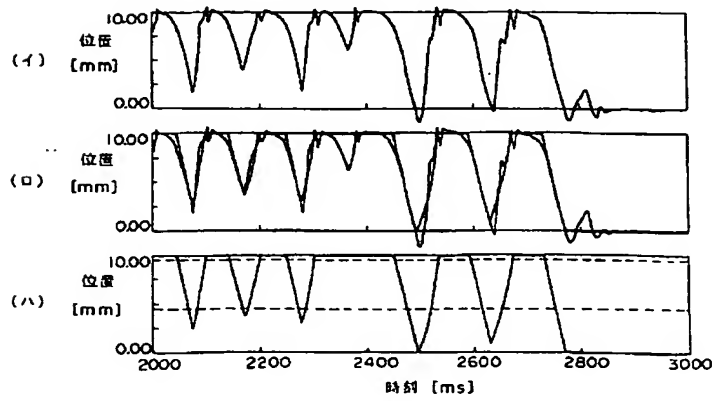
【図16】



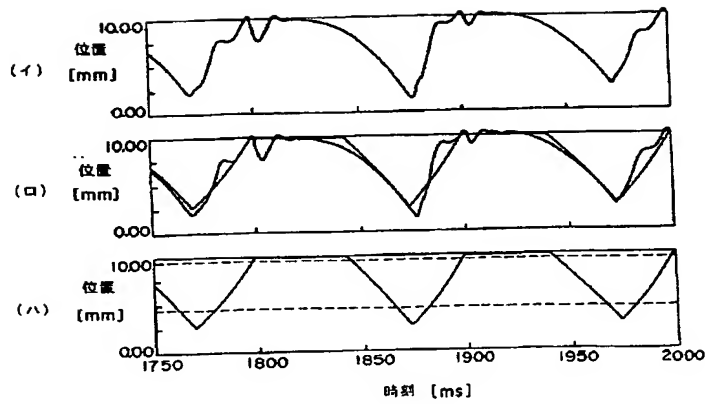
【図17】



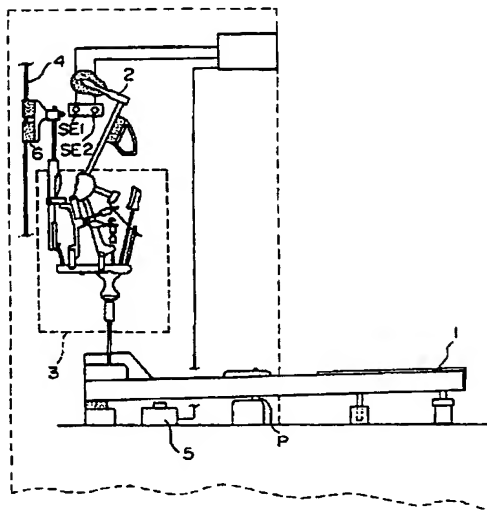
【図18】



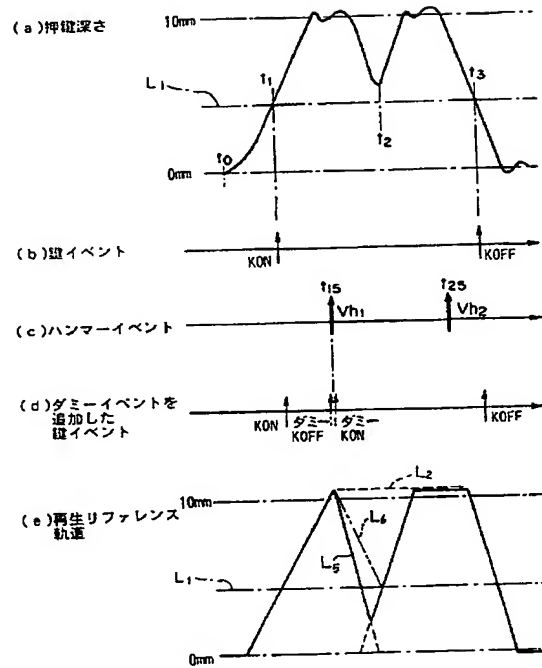
【図19】



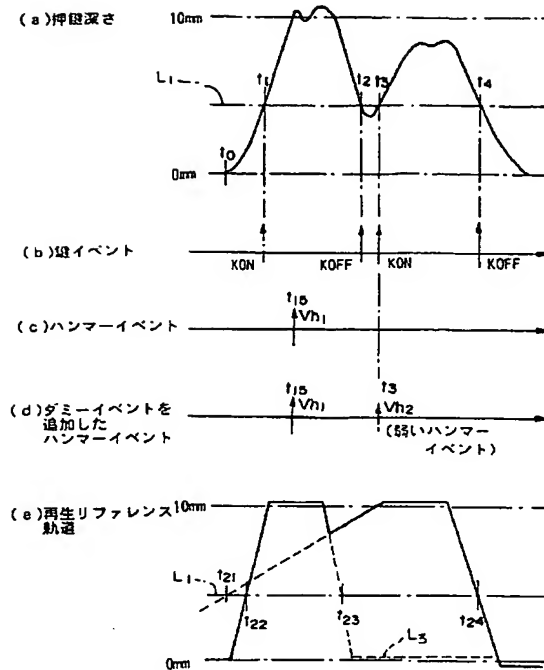
【図20】



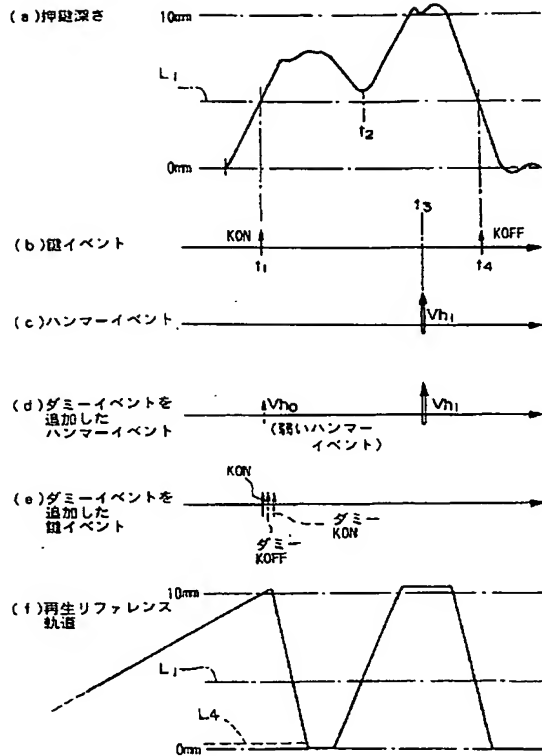
【図21】



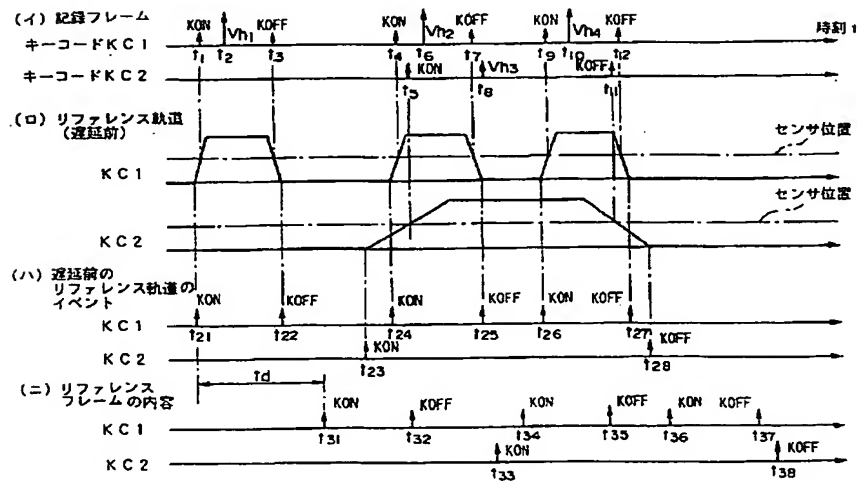
【図 22】



【図 23】

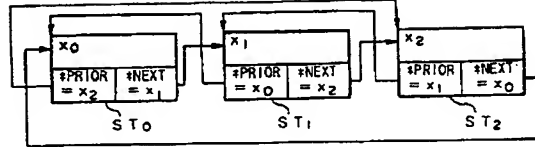
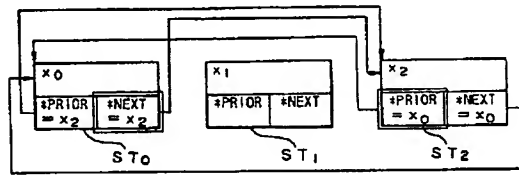
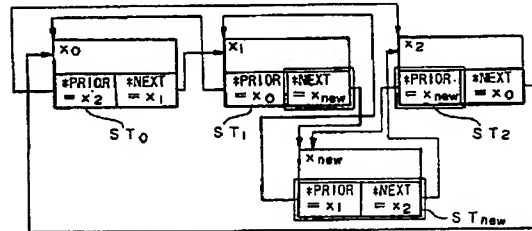


【図 26】



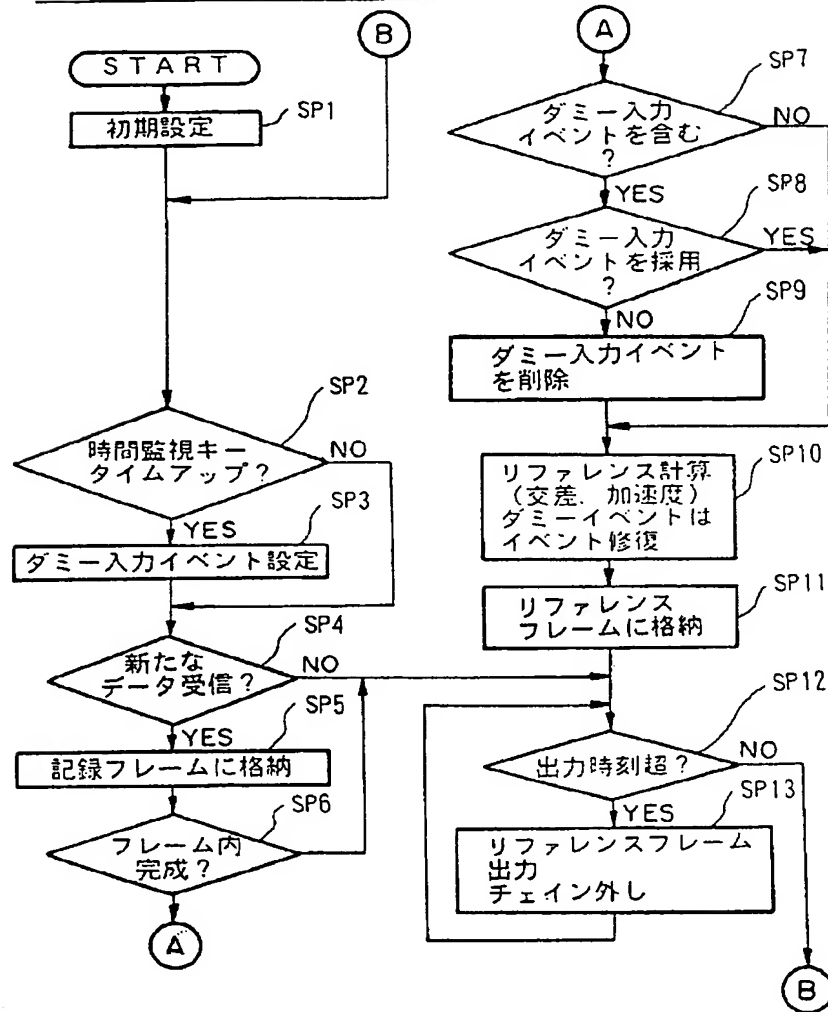
【図24】

(イ) 変更前

(ロ) ST₁の削除(ハ) ST_{new}の追加

【図25】

リアルタイム処理装置300の動作フローチャート



フロントページの続き

(72)発明者 大場 保彦
 静岡県浜松市中沢町10番1号 ヤマハ株式
 会社内

特開平7-271355

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第2区分

【発行日】平成13年8月31日(2001.8.31)

【公開番号】特開平7-271355

【公開日】平成7年10月20日(1995.10.20)

【年通号数】公開特許公報7-2714

【出願番号】特願平6-79604

【国際特許分類第7版】

G10F 1/02

【F1】

G10F 1/02 B

【手続補正書】

【提出日】平成12年10月26日(2000.10.26)

【手続補正1】

【補正対象書類名】図面

【補正対象項目名】図5

【補正方法】変更

【補正内容】

【図5】

